National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

ROBUSTNESS ISSUES IN LONG-RANGE PREDICTIVE CONTROL

BY

©

PRANOB BANERJEE

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

DOCTOR OF PHILOSOPHY

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1996

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-612-10570-9

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Pranob Banerjee

TITLE OF THESIS: Robustness Issues In Long-Range Predictive Control

DEGREE: Doctor of Philosophy

YEAR THIS DEGREE GRANTED: 1996

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

ADDRESS:

Chinchbhawan, Wardha Road, PO Khapri,
Nagpur, 441108,
India.

Date: January 31, 1996

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled

Robustness Issues In Long-Range Predictive Control

submitted by Pranob Banerjee in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Process Control.

Dr. S.L. Shah
(Supervisor)

DG Fisher
Dr. D.G. Fisher
(Chemical Engineering)

Dr. R.K. Wood
(Chemical Engineering)

Keith Stromsmoe
Dr. K. Stromsmoe
(Electrical Engineering)

Dr. D.E. Rivera
(Chemical Engineering, Arizona State University, Tempe, AZ, USA)

Date: January 29, 1996

॥ ॐ नमो नारायणाय ॥

Dedicated To
My Parents

ABSTRACT

A long range predictive control (LRPC) law is designed on the assumption, that the given model truly reflects the plant dynamics. However in reality there is always a model-plant mismatch (MPM) which gives rise to stability and performance problems when the designed LRPC is implemented on the actual plant. This thesis addresses the crucial question of designing a LRPC that is robust to MPM and looks at ways to enhance the actual performance of the LRPC control loop in the presence of MPM.

The small gain theorem is used as a tool to design a LRPC that is robust to MPM. Such a robust design method requires the knowledge of the MPM, which can be estimated from the plant data using signal processing methods. This robust design method has been applied to generalized predictive control (GPC) and Markov-Laguerre based model predictive control (MPC) laws. It is shown that, irrespective of the model types, the robustness bounds of both these LRPCs behave similarly under the influence of tuning parameters. In the case of GPC: (a) the robustness properties associated with important tuning parameters are established analytically and verified experimentally; (b) it is shown that the model and MPM can be estimated from closed loop data; and (c) an optimization problem is formulated within the small gain framework to select some of the controller tuning parameters. For the Markov-Laguerre MPC: (a) designed stability is improved by incorporating steady state weighting; and (b) faster disturbance rejection is obtained by including a structured noise model in the controller design.

Ideally, a stable LRPC with satisfactory performance can be obtained by estimating a model with minimal MPM. Therefore significant emphasis has been given to system identification methods and their applications have been illustrated through two industrial case studies. In the context of model estimation, an extension based on the augmented UD identification method to simultaneously estimate parameters of different orders of the orthonormal function models is developed.

Three control relevant identification methods are reviewed and suitably modified to

enhance the achieved performance of GPC. The key issue in control relevant identification is to bring the designed and the achieved closed loop performances as close as possible. This is accomplished by: (a) designing a suitable model estimation filter; and (b) appropriately modifying the designed controller objective function, and using them to obtain control relevant models and thereby upgrading the performance.

ACKNOWLEDGEMENT

This thesis is a result of contributions in numerous forms from several persons over the past few years. I am thankful to Dr. S. L. Shah for his excellent supervision and guidence on all aspects of this thesis. I am grateful to Dr. Shah for his timely advice on several occasions. I am grateful to Dr. D. G. Fisher for his overall guidence and notably for his suggestions on system identification and robustness issues. I am thankful to Dr. R. K. Wood for his support and discussing the optimization problem. I thank my external examiner Dr. D. E. Rivera for agreeing to come over to frigid Alberta (-33°C) from balmy Arizona (22°C) in the middle of winter.

Thanks are due to my colleagues in the process control group: Dr. Danyang Liu, Dr. Lanre Badmus, Dr. Munawar Saudagar, Dr. Fengxi Zhou, Dr. Ezra Kwok (the γ_∞ ~~guy~~ prof.), Dr. Steve Niu (the $\mathcal{AUDI}$ doc.), Kent Qi, Randy Miller and S. Lakshminarayanan (Laks), for their technical help on many matters; Dr. Ravindra Gudi for his overall help and for sharing his musical insights; Biao Huang for sharing his thoughts on performance issues and for those naughty mao and hauza comments; Dr. Mary Bourke (holy ~~cow~~ cat!), Dr. Viral Maniar, Roman Walther, Garry Kwong, Sreekanth Lalgudi (the guy loaded with witty remarks), Albert Chiu, Chee Wong, Ricky Leung, Amy, Rohit Patwardhan and V.Anand, for their lively 'technical' discussions. My special thanks are also due to Laks and Rohit for helping me in implementing the corrections.

I thank: Anand R. Kalvey, for sharing his expertise in signal processing and photography; Narendra Ravi, for his help; Ajay Goyal, R. Rangiya and Nilesh Oak for sharing their philosophical insights; Arun Bhargava, K.R.Prasad, Sanjay Madan, Chiradeep Vittal and others for making my stay in Edmonton enjoyable.

I thank the Commonwealth Scholarship and Fellowship program of Canada and the NSERC research grant for their financial support.

My deep thanks are due to Ruma my wife and my little daughter Amrita for their perseverance and shouldering the responsibilities. I thank my brother Lt. Rajshekhar Banerjee and his wife Chandra for constantly keeping in touch with me from India.

I am unable to find enough words to express my thanks and gratitude to my parents, hence I dedicate this thesis to them.

Contents

1 Introduction **1**

1.1 Long range predictive control 1

1.2 Motivation and objective 3

1.3 Scope 5

1.4 Structure 6

2 Signal Processing for System Identification **11**

2.1 Introduction 11

2.2 Signal processing - a historical account[2] 13

2.3 Signal characterization 15

2.4 Signal processing in time domain 18

2.4.1 Autocovariance/Autocorrelation 18

2.4.2 Cross covariance/cross correlation 19

2.4.3 Applications of acf 20

2.4.4 Applications of ccf 25

2.5 Signal processing in frequency domain 27

2.5.1 Continuous time periodic signal 27

2.5.2 Non-periodic continuous signal 28

2.5.3 Non-periodic discrete signal 29

2.5.4 Discrete time periodic signal 30

2.5.5 Implementation of DFT (FFT) 31

2.5.6 Data windowing 33

2.5.7 Power spectrum 35

2.5.8 Spectral smoothing 37

2.5.9 Estimation of frequency response of a transfer function 39

2.6 Statistical analysis in signal processing 39
  2.6.1 Confidence bounds on smoothened auto spectrums 41
  2.6.2 Confidence bounds on the smoothened cross spectrum 43
  2.6.3 Confidence bound on smoothened gain and phase spectrums 44
2.7 Case studies 46
  2.7.1 Case: Noise free plant 47
  2.7.2 Case: Data corrupted by white measurement noise 49
  2.7.3 Case: Data corrupted by ARMA noise 55
2.8 Estimation of model-plant mismatch (MPM) 58
  2.8.1 Spectral estimation of MPM - simulation example 61
2.9 Conclusions 65

3 System Identification - Case Studies **73**
3.1 Introduction 73
3.2 Modelling linear systems 75
3.3 Frequency response models 76
  3.3.1 Estimation of frequency response models 77
3.4 Step response and FIR models 77
  3.4.1 Estimation of Step/FIR models 78
3.5 Transfer function models 78
  3.5.1 Estimation of transfer function models 79
3.6 Orthonormal function models 81
  3.6.1 Estimation of Markov-Laguerre models 83
3.7 State space models 83
3.8 Confidence bounds 83
3.9 Steps in model identification and validation 85
3.10 Case-I: Shell benchmark problem 86
  3.10.1 Problem description 87
  3.10.2 Overall solution procedure 88
  3.10.3 Identification of noise models 89
  3.10.4 Plant dynamic data and preliminary analysis 93
  3.10.5 Model estimation 98
  3.10.6 Model validation 101

    3.10.7 Residual analysis 108
    3.10.8 Step responses 110
  3.11 Case-II: An industrial plant 114
    3.11.1 Preliminary analysis 116
    3.11.2 Estimation of models 116
    3.11.3 Model validation 118
    3.11.4 Residual analysis 118
    3.11.5 Step responses 118
  3.12 Conclusions 123

4 Parameter Estimation for Orthonormal Function Models using AUDI 127
  4.1 Introduction 127
  4.2 Historical perspective 130
  4.3 Definitions 132
  4.4 AUDI formulation of orthonormal function models 137
    4.4.1 Data regressor and parameters for the AUDI method 137
    4.4.2 LS properties of the AUDI estimated model parameters $\xi(t, N)$ . . 139
    4.4.3 AUDI formulation for the orthonormal series models 140
    4.4.4 Effect of noise and modelling error in the AUDI method 143
  4.5 Simultaneous estimation of Laguerre models with real poles 145
    4.5.1 Simulation Example 146
  4.6 Simultaneous estimation of Laguerre models with complex poles 149
  4.7 Simultaneous estimation of Kautz models 149
    4.7.1 Simulation Example 150
  4.8 Simultaneous estimation of FIR Models 150
    4.8.1 Simulation Example 152
  4.9 Application of the AUDI method for estimation of Markov-Laguerre models 154
    4.9.1 Simulation Example 154
  4.10 Conclusions 155

5 Robust Design of Generalized Predictive Control (GPC). 161
  5.1 Introduction 161
  5.2 Generalized predictive Control 163

5.3 Diophantine equations 165

5.3.1 The quotient polynomial $E_i(q)$ in the Diophantine identity 167

5.3.2 The remainder polynomial $F_i(q)$ in the Diophantine identity 167

5.3.3 The quotient polynomial $\mathcal{G}_i(q)$ in the Diophantine identity 168

5.3.4 The remainder polynomial $\bar{G}_i(q)$ in the Diophantine identity 169

5.4 Linear form of GPC 169

5.4.1 The control law matrix h 170

5.4.2 Linear GPC polynomial $S(q)$ 171

5.4.3 Linear GPC polynomial $T(q)$ 171

5.5 Robust design of GPC 172

5.6 Robustness of GPC tuning parameters 176

5.7 Effect of N_2 180

5.8 Effect of λ 186

5.9 Effect of $C_c(q)$ 190

5.10 Effect of NU 192

5.11 Effect of $P(q)$ 194

5.12 Effect of γ-weighting 194

5.13 Effect of model 195

5.14 Robustness of GPC in presence of disturbances 196

5.15 Experimental evaluation 197

5.16 Estimation of MPM from the Closed-loop data 199

5.17 Selection of tuning parameters using an optimization technique 205

5.18 Conclusions 208

6 Robustness of Markov-Laguerre model based Predictive Controller 214

6.1 Introduction 214

6.2 Markov-Laguerre based state-space model 216

6.3 Design of unconstrained MPC - without noise model 218

6.3.1 Steady State Weighting - γ_∞ 220

6.4 Design of constrained MPC 222

6.5 Design of unconstrained MPC - with a noise model 224

6.6 Robustness Analysis 229

6.7 Simulation Results 234

6.7.1 Effect of N_2 235

6.7.2 Effect of λ 236

6.7.3 Effect of NU 237

6.7.4 Effect of γ_∞ 237

6.7.5 Comparison between GPC and Markov-Laguerrre MPC 238

6.7.6 Effect of noise model on the performance of Markov-Laguerre MPC 240

6.8 Conclusions 242

7 Control Relevant Identification for GPC **247**

7.1 Introduction 247

7.2 Problem formulation 250

7.2.1 Performance criteria for GPC 253

7.3 Control-relevant identification by Shook and Shah 255

7.4 Control-relevant identification by Rivera *et al.* 259

7.5 Control-relevant identification by Zang *et al.* 261

7.5.1 Weighted objective function and iterative design 265

7.6 A GPC algorithm with frequency weightings 267

7.6.1 Iterative implementation of L_s 269

7.6.2 Iterative implementation of L_r 273

7.6.3 Iterative implementation of L_z 277

7.7 Conclusions 282

8 Conclusions, recomendations and research directions **286**

8.1 Concluding remarks 286

8.1.1 Main contributions 286

8.1.2 5. Control relevant identification for GPC 288

8.2 General design and tuning guidelines for LRPC 289

8.3 Future research directions 289

List of Tables

2.1 Time and frequency representation of different windows 70

3.1 Process objectives, operating conditions and constraints 88
3.2 Estimated models for P/D for different SNRs. 100
3.3 Estimated models for P/Q for different SNRs. 100
3.4 Estimated models for X/Q for different SNRs. 101

4.1 Classical Orthonormal Functions and their relevant intervals. 130
4.2 Laguerre gains obtained using LS_y method. 158
4.3 Laguerre gains obtained using AUDI method. 158
4.4 Comparison of FIR coefficients obtained using LS_y and AUDI methods. . . 158

5.1 Different cases of GPC parameters for disturbance rejection. 210
5.2 Different cases of GPC parameters for the experimental set-up. 210
5.3 Layout of different cases studies to determine optimal tuning parameters. . 210
5.4 Small gain theorem based optimal tuning parameters of GPC. 210

List of Figures

2.1 Block diagram of plant and model. 16

2.2 Top: White noise signal. Bottom: Autocorrelation of white noise signal. . . 21

2.3 Top: MA(2) signal. Bottom: Autocorrelation of MA(2) signal. 22

2.4 Top: AR(1) signal. Bottom-left: Autocorrelation of AR(1) signal. Bottom-right: Partial autocorrelation of the same signal. 24

2.5 Application of cross correlation function to estimate the process characteristics. 26

2.6 Top: Different window functions in time domain. Bottom-left: Amplitude of different windows in frequency domain. Bottom-right: Magnitude of these window functions in frequency domain. 34

2.7 Use of periodogram to estimate the variance of a white noise signal. 42

2.8 Estimation of Bode plots for a noise free plant using (i) white noise excitation and (ii) no spectral smoothing. Left: Magnitude spectrum. Right: Phase spectrum. 47

2.9 Estimation of Bode plots for a noise free plant with bounds using (i) white noise excitation and (ii) 1 Hanning window of length $M = 1024$. Left: Magnitude spectrum. Right: Phase spectrum. 48

2.10 Estimation of Bode plots for a noise free plant with upper and lower bounds using (i) white noise excitation and (ii) 4 Hanning windows of length $M = 256$ each. Left: Magnitude spectrum. Right: Phase spectrum. 48

2.11 Estimation of Bode plots for a noise free plant with upper and lower bounds using (i) white noise excitation and (ii) 4 Hanning windows of length $M = 1024$ each. Left: Magnitude spectrum. Right: Phase spectrum. 49

2.12 Top: Noise free process output for a square wave input. Bottom: Estimation of Bode plots and their bounds for the above case with 4 Hanning windows of length $M = 256$ each. Left: Magnitude spectrum. Right: Phase spectrum. 50

2.13 Top: Time series plots of noisy ($y(t)$) and noise free ($y^*(t)$) process outputs for a white noise excitation. $v(t)$ is the white measurement noise sequence (SNR=3.03). Bottom: Estimation of Bode plots for the above case with no spectral smoothing. Bottom-Left: Magnitude spectrum. Bottom-Right: Phase spectrum. 51

2.14 Estimation of Bode plots and their bounds for a plant corrupted by white noise (SNR=3.03) using (i) white noise excitation and (ii) 1 Hanning window of length $M = 1024$. Left: Magnitude spectrum. Right: Phase spectrum. . . 52

2.15 Estimation of Bode plots, their bounds and coherency spectrum for a plant corrupted by white noise (SNR=3.03) using (i) white noise excitation and (ii) 4 Hanning windows of length $M = 256$ each. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Squared coherency spectrum. 53

2.16 Estimation of Bode plots, their bounds and coherency spectrum for a plant corrupted by white noise (SNR=3.03) using (i) white noise excitation and (ii) Bartlett's smoothing method. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Squared coherency spectrum. 54

2.17 Effect of level of white measurement noise (SNR) on the squared coherency spectrum. 55

2.18 Top: Time series plots of noise free output ($y^*(t)$) and output corrupted by ARMA noise ($y(t)$) for white noise excitation. $v(t)$ is the ARMA noise sequence (SNR$\approx$ 3.0). Bottom: Magnitude spectrum of $v(t)$. 56

2.19 Estimates of Bode plots and coherency spectrum for a plant corrupted by ARMA noise (SNR$\approx$ 3.0) using (i) white noise excitation and (ii) 16 Hanning windows of length $M = 256$ each. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Square coherency spectrum. 57

2.20 Top: Time series plots of low pass filtered white noise input $u^f(t)$ and its output $y(t)$ that is corrupted by ARMA noise (SNR$\approx$ 3.0). Bottom: Magnitude spectrum of filtered input. 58

2.21 Estimation of Bode plots and coherency spectrum for a plant corrupted by ARMA noise (SNR=3.03) using (i) low pass filtered white noise input and (ii) 16 Hanning windows of length $M = 256$ each. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Square coherency spectrum. 59

2.22 Nyquist plots of the plant, model and the uncertainty regions. 62

2.23 Left: Spectral estimation of MPM with square wave input for a noise free plant. Right: Error between true and estimated MPM. 63

2.24 Left: Spectral estimation of MPM with white noise input for a noise free plant. Right: Error between true and estimated MPM. 63

2.25 Spectral estimation of MPM in the presence of white measurement noise (SNR$\approx$ 1.4). Left: with square wave input, Right: with white noise input. . 64

3.1 Noise in P and its ACFs for different SNRs. Top: Time series data. Bottom: ACF. 90

3.2 Noise in X and its ACFs for different SNRs. Top: Time series data. Bottom: ACF. 91

3.3 ACFs and PACFs of differenced noise data for different SNRs. Top-Left and Right: ACF of ΔP and ΔX. Bottom Left and Right: PACF of ΔP and ΔX. 92

3.4 Plant responses for different SNRs when subjected to excitation in D. . . . 94

3.5 Plant responses for different SNRs when subjected to excitation in Q. . . . 95

3.6 Squared coherency functions for different channels and different SNRs. Top-Left: Channel P/D. Top-Right: Channel X/D. Bottom-Left: Channel P/Q. Bottom-Right: Channel X/Q. 96

3.7 Cross correlation functions for different channels and different SNRs. Top-Left: Channel P/D. Top-Right: Channel X/D. Bottom-Left: Channel P/Q. Bottom-Right: Channel X/Q. 97

3.8 Frequency domain model validation for channel P/D. Top-Left: Magnitude spectrum for 20% noise. Top-Right: Phase spectrum for 20% noise. Bottom-Left: Magnitude spectrum for 100% noise. Bottom-Right: Phase spectrum for 100% noise. 102

3.9 Frequency domain model validation for first order models for channel P/Q. Top-Left: Magnitude spectrum for 20% noise. Top-Right: Phase spectrum for 20% noise. Bottom-Left: Magnitude spectrum for 100% noise. Bottom-Right: Phase spectrum for 100% noise. 103

3.10 Frequency domain model validation for channel X/Q. Top-Left: Magnitude spectrum for 20% noise. Top-Right: Phase spectrum for 20% noise. Bottom-Left: Magnitude spectrum for 100% noise. Bottom-Right: Phase spectrum for 100% noise. 104

3.11 Time domain model validation for channel P/D. Top-Left: Model output with feedback for 20% noise. Top-Right: Model output without feedback for 20% noise. Bottom-Left: Model output with feedback for 100% noise. Bottom-Right: Model output without feedback for 100% noise. 105

3.12 Time domain model validation for channel P/Q. Top-Left: Model output with feedback for 20% noise. Top-Right: Model output without feedback for 20% noise. Bottom-Left: Model output with feedback for 100% noise. Bottom-Right: Model output without feedback for 100% noise. 106

3.13 Time domain model validation for channel X/Q. Top-Left: Model output with feedback for 20% noise. Top-Right: Model output without feedback for 20% noise. Bottom-Left: Model output with feedback for 100% noise. Bottom-Right: Model output without feedback for 100% noise. 107

3.14 Residual analysis for channel P/D. Top-Left: ACF(Δe) for 20% noise. Top-Right: ACF($\hat{D}_1(q)\Delta e$) for 20% noise. Bottom-Left: ACF(Δe) for 100% noise. Bottom-Right: ACF($\hat{D}_2(q)\Delta e$) for 100% noise. 108

3.15 Residual analysis for channel P/Q. Top-Left: ACF(Δe) for 20% noise. Top-Right: ACF($\hat{D}_1(q)\Delta e$) for 20% noise. Bottom-Left: ACF(Δe) for 100% noise. Bottom-Right: ACF($\hat{D}_2(q)\Delta e$) for 100% noise. 109

3.16 Residual analysis for channel X/Q. Top-Left: ACF(Δe) for 20% noise. Top-Right: ACF($\hat{D}_1(q)\Delta e$) for 20% noise. Bottom-Left: ACF(Δe) for 100% noise. Bottom-Right: ACF($\hat{D}_2(q)\Delta e$) for 100% noise. 110

3.17 Step responses and MPM for different identified models for channel P/D. Top-Left: Step responses of models identified for 20% noise. Top-Right: Step responses of models identified for 100% noise. Bottom-Left: MPM for models identified for 20% noise. Bottom-Right: MPM for models identified for 100% noise. 111

3.18 Step responses and MPM for different identified models for channel P/Q. Top-Left: Step responses of models identified for 20% noise. Top-Right: Step responses of models identified for 100% noise. Bottom-Left: MPM for models identified for 20% noise. Bottom-Right: MPM for models identified for 100% noise. 112

3.19 Step responses and MPM for different identified models for channel X/Q. Top-Left: Step responses of models identified for 20% noise. Top-Right: Step responses of models identified for 100% noise. Bottom-Left: MPM for models identified for 20% noise. Bottom-Right: MPM for models identified for 100% noise. 113

3.20 Process output and input data. 114

3.21 Preliminary data analysis. Top: Squared coherency function. Bottom: Cross correlation function. 115

3.22 Different error analyis. (a) Delay estimation using AUDI for ARX model structure. (b) Effect of delay on the loss functions of ARMAX(3,3,1) model for data set 1, 2 and 3. (c) Effect of delay on $(|G_\phi(\omega)| - |\hat{G}(\omega)|)^2$. (d) Effect of delay on $(\text{ang}G_\phi(\omega) - \text{ang}\hat{G}(\omega))^2$. 116

3.23 Frequency domain model validation. Top: Magnitude spectrum. Bottom: Phase spectrum. 119

3.24 Time domain model validation. (a) Model output for data set-1. (b) Model output for data set-2. (c) Model output for data set-3. (d) Model output with feedback for data set-1. 120

3.25 Residual analysis for ARMAX model. Top-Left: Filtered residual data. Top-Right: ACF of filtered residual. Bottom-Left: PACF of filtered residual. Bottom-Right: MPM. 121

3.26 Residual analysis for Markov-Laguerre model. Top-Left: Differenced residual data. Top-Right: ACF of differenced residual. Bottom-Left: PACF of differenced residual. Bottom-Right: MPM. 122

3.27 Step responses for different identified models. 123

4.1 Comparison between AUDI and LS_y methods for 2^{nd} and 3^{rd} order Laguerre models. 147

4.2 Magnitude Spectrum of the modelling errors for the AUDI and LS_y based Laguerre models for orders 2 to 4. 147

4.3 Effect of noise in the estimation of Laguerre model via the AUDI method . 148

4.4 Comparison between AUDI and LS_y methods for 3^{rd} order Laguerre models for noisy data. 148

4.5 Comparison between AUDI and LS_y methods for 4^{th} and 8^{th} order Kautz and Lag_c models. 150

4.6 Parameter profile for 8^{th} order Kautz and Lag_c models obtained via the AUDI and LS_y methods. 150

4.7 Comparison between AUDI and LS_y methods for 8^{th} order Kautz/Lag_c models for noisy data. 151

4.8 A comparison between 15^{th} and 30^{th} order FIR coefficients obtained using AUDI and LS_y methods. 152

4.9 A comparison between 15^{th} and 30^{th} order FIR models obtained using AUDI and LS_y methods. 152

4.10 A comparison of 30^{th} order step response coefficients obtained using the AUDI and LS_y methods. 153

4.11 A comparison between 3^{rd} and 4^{th} order Markov-Laguerre models obtained using the AUDI and LS_y methods. 154

5.1 $\mathcal{S}_1 - \mathcal{S}_2$ structure for the Small Gain Theorem. 172

5.2 Feedback structure of GPC. 172

5.3 Servo responses for tuning sets 1 and 2. 178

5.4 Robustness bounds or robustness margins for tuning sets 1 and 2. 178

5.5 Nyquist plots of $\hat{G}(\omega)$, $\mathcal{C}_1(\omega)$ and $\mathcal{C}_2(\omega)$. 179

5.6 Effect of N_2 on the robustness bound. 181

5.7 Effect of N_2 on the servo tracking. 181

5.8 Effect of N_2 on the magnitude spectrum of $\mathcal{C}(\omega)$. 183

5.9 Effect of N_2 on the phase spectrum of $\mathcal{C}(\omega)$. 183

5.10 Plot of $\beta = |\mathcal{C}(\omega)|_{N_2}/|\mathcal{C}(\omega)|_{N_2+1}$ and $\beta^\star = |\mathcal{C}^\star(\omega)|_{N_2}/|\mathcal{C}(\omega)|_{N_2+1}$. 185

5.11 Effect of λ on the robustness bound. 186

5.12 Effect of λ on the servo tracking. 186

5.13 Effect of λ on the magnitude spectrum of $\mathcal{C}(\omega)$. 188

5.14 Effect of λ on the phase of $\mathcal{C}(\omega)$. 188
5.15 Plot of $\beta = |\mathcal{C}(\omega)|_{\lambda_1}/|\mathcal{C}(\omega)|_{\lambda_2}$ and $\beta^\star = |\mathcal{C}^\star(\omega)|_{\lambda_1}/|\mathcal{C}(\omega)|_{\lambda_2}$ 189
5.16 Effect of $C_c(q)$ on the robustness bound. 190
5.17 Effect of $C_c(q)$ on the servo tracking. 190
5.18 Effect of $C_c(q)$ on the magnitude spectrum of $\mathcal{C}(\omega)$. 191
5.19 Effect of $C_c(q)$ on the phase spectrum of $\mathcal{C}(\omega)$. 191
5.20 Effect of NU on the robustness bound. 193
5.21 Effect of NU on the servo tracking. 193
5.22 Effect of $P(q)$ on the robustness bound. 194
5.23 Effect of $P(q)$ on the servo tracking. 194
5.24 Effect of γ_{inf} on the robustness bound. 195
5.25 Effect of γ_{inf} on the servo tracking. 195
5.26 Effect of model on the robustness bound. 196
5.27 Effect of model on the servo tracking. 196
5.28 Effect of tuning parameters on the robustness bound in the presence of measurement noise and disturbances. 197
5.29 Effect of parameters on the servo tracking in the presence of measurement noise and disturbances. 197
5.30 Experimental set-up. 198
5.31 Plant input-output data. 198
5.32 Effect of tuning parameters on the robustness bound for the experimental set-up. 199
5.33 Effect of tuning parameters on the servo tracking for the experimental set-up. 199
5.34 Noise-free closed loop output data for step changes in the setpoint. 203
5.35 Estimated MPM from the noise-free closed loop data. 203
5.36 A section of closed loop data corrupted with white measurement noise for white noise excitation in the setpoint. 203
5.37 Estimated MPM from the noise corrupted closed loop data. 203
5.38 Estimated model from the noise-free closed loop data. 204
5.39 Estimated model from the closed loop data corrupted by the measurement noise. 204
5.40 Robustness bounds for three optimal sets of tuning parameters for $\epsilon = 0.1$. . 205

5.41 Servo responses for three optimal sets of tuning parameters for $\epsilon = 0.1$. . . . 205

5.42 Robustness bounds for three optimal sets of tuning parameters for $\epsilon = 0.2$. . 207

5.43 Servo responses for three optimal sets of tuning parameters for $\epsilon = 0.2$. . . . 207

6.1 Closed loop structure of Markov-Laguerre Predictive Controller 230

6.2 Small Gain Structure for the Markov-Laguerre Predictive Controller 230

6.3 Effect of N_2 on the robustness margin. 235

6.4 Effect of N_2 on the servo and regulatory response. 235

6.5 Effect of λ on the robustness margin. 236

6.6 Effect of λ on the servo and regulatory response. 236

6.7 Effect of NU on the robustness margin. 237

6.8 Effect of NU on the servo and regulatory response. 237

6.9 Effect of γ_∞ on the robustness margin. 238

6.10 Effect of γ_∞ on the servo and regulatory response. 238

6.11 Robustness margins for GPC and Markov-Laguerre based MPC for a noise-free plant. 239

6.12 A comparative performance between GPC and Markov-Laguerre based MPC for a noise-free plant. 239

6.13 Robustness margins for GPC and Markov-Laguerre based MPC for a noisy plant. 240

6.14 A comparative performance between GPC and Markov-Laguerre based MPC for a noisy plant. 240

6.15 Performance of constrained GPC and Markov-Laguerre based MPC for a noisy plant. 241

6.16 Input constraint stabilizes the system but the input hits the saturation limits more often. 241

6.17 Robustness margins for GPC and Markov-Laguerre based MPC for a noisy plant. 241

6.18 A comparative performance between GPC and Markov-Laguerre based MPC for a noisy plant. 241

6.19 Robustness margins for Markov-Laguerre based MPCs with and without the noise model. 242

6.20 Regulatory behavior of the Markov-Laguerre based MPCs with and without the noise model. 242
6.21 Robustness margins for GPC and Markov-Laguerre based MPC with a noise model. 243
6.22 Performance of GPC and Markov-Laguerre based MPC with a noise model. 243

7.1 Designed closed-loop system. 250
7.2 Achieved closed-loop system. 250
7.3 Implementation of LRPI. 258
7.4 A comparison of the robustness margins for the GPC corresponding to the models obtained using the LS and the LRPI methods. 258
7.5 Effect of L_s on performance of the system 259
7.6 Effect of L_r on performance of the system 261
7.7 Designed closed loop system. 261
7.8 Achieved closed loop system. 261
7.9 Bode plots for the L_s, L_r and the L_z filters. 264
7.10 Effect of L_z on performance of the system 264
7.11 A comparison of the robustness margins for GPC due to L_s, L_r and L_z filters.264
7.12 A comparison of GPC servo responses due to L_s, L_r and L_z filters. 264
7.13 Iterative control-relevant design method. 266
7.14 The influence of L_s on the progress of designed-vs-achieved performance criteria along the iteration steps. 270
7.15 The effect of L_s on the achieved performance at iterations 1 and 6. 270
7.16 The effect of frequency weightings F_1 and F_2 on the achieved performance with the use of L_s-filter at iteration-6. 271
7.17 Evolution of the estimated models by using L_s. 271
7.18 Evolution of $\Delta L_s/C_c$ with the progress in iteration. 271
7.19 The frequency weighting functions F_1 and F_2 at iteration 1 and the robustness margins at iterations 1 and 2 with the use of L_s-filter. 272
7.20 The frequency weighting functions F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6 with the use of L_s-filter. 272
7.21 The evolution F_1 for the case of L_s-filter. 273
7.22 The evolution of F_2 for the case of L_s-filter. 273

7.23 The influence of L_r on the progress of designed-vs-achieved performance criteria along the iteration steps. 274

7.24 The effect of L_r on the achieved performance at iterations 1, 3 and 6. . . . 274

7.25 The effect of frequency weightings on the achieved performance with the use of L_r-filter at iteration 6. 274

7.26 Evolution in the estimated models by using L_r. 275

7.27 Evolution of L_r with the progress in iteration. 275

7.28 The frequency weighting filters F_1 and F_2 at iteration 1 and the robustness margins at iterations 1 and 2 for the case of L_r-filter. 276

7.29 The frequency weighting filters F_1 and F_2 at iteration 2 and the robustness margins at iterations 2 and 3 for the case of L_r-filter. 276

7.30 The frequency weighting filters F_1 and F_2 at iteration 3 and the robustness margins at iterations 3 and 4 for the case of L_r-filter. 277

7.31 The frequency weighting filters F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6 for the case of L_r-filter. 277

7.32 The evolution F_1 for the case of L_r-filter. 278

7.33 The evolution of F_2 for the case of L_r-filter. 278

7.34 The influence of L_z on the progress of designed-vs-achieved performance criteria along the iteration steps. 279

7.35 Achieved performances at different iterations for the case of L_z-filter. 279

7.36 The effect of frequency weightings on the achieved performance with the use of L_z-filter at iteration-6. 280

7.37 The effect of L_z on the evolution of model. 280

7.38 The evolution of L_z along the iteration steps. 280

7.39 The frequency weighting filters F_1 and F_2 at iteration 1 and the robustness margins at iterations 1 and 2 for the case of L_z-filter. 281

7.40 The frequency weighting filters F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6 for the case of L_z-filter. 281

7.41 The evolution F_1 for the case of L_z-filter. 282

7.42 The evolution of F_2 for the case of L_z-filter. 282

7.43 A comparison of different achieved performances obtained using different control-relevant filters. 283

7.44 A comparison of servo responses obtained using different control-relevant filters. 283

Chapter 1

Introduction

1.1 Long range predictive control

The term *long range predictive control* (LRPC) is used generically to represent a family of model based predictive controllers. All LRPC strategies use a model based description of the plant dynamics to determine the controller moves. The LRPC algorithms in general calculate several controller moves in the future, but they implement only the first control action and hence such controllers are also called *receding horizon* controllers. The LRPC design is based on the assumption that the given mathematical model truly reflects the plant dynamics; therefore such controllers are also categorized as *certainty equivalence controllers*.

There are varied opinions on how the concept of LRPC emerged. Clarke and Mohtadi who developed one popular LRPC algorithm, attribute the earliest LRPC design to Dawkins and Briggs in 1965 [1, 2]. Another reference to the LRPC ideas can be traced back to Kishi, in 1964 [3]. Kishi's work on model based control was motivated from several previous results on optimal and adaptive control due to Kalman in 1958, Merriam in 1960 and others [4, 5, 3]. Kishi even refers to a paper by Horing, who explicitely used the term *predictive control* in his work in 1962 [3, 6]. Nevertheless, according to most references on LRPC, the present state-of-the-art in the design of LRPC is said to conceptually stem from the *identification and control* algorithm (IDCOM) and *dynamic matrix control* (DMC) algorithms which were respectively proposed by Richalet *et al.* in 1978 [7] and Cutler and Ramaker in 1980 [8]. We thus see that the concept of LRPC that was presumably conceived during late 1970s and early 1980s was already examined in the early to mid 1960s.

The interest in LRPC surged since the advent of IDCOM and DMC because of their successful industrial applications. The research interest in LRPC was at its peak during the decade of 1980s when scores of several other LRPC algorithms were proposed. Some of the LRPC controllers that emerged in the 1980s and that are traditionally quoted in the LRPC literature are: *model algorithmic control* (MAC) due to Rouhani and Mehra in 1982 [9], *internal model control* (IMC) due to Garcia and Morari in 1982, *optimal control syntheisis*

(OCM) due to Peterka in 1982 [10, 11], *extended horizon adaptive control* (EHAC) due to Ydstie in 1984 [12], *multipredictor receding horizon adaptive control* (MURHAC) due to Lemos and Musca in 1985 [13], *multistep multivariable adaptive controller* (MUSMAR) due to Greco *et al.* in 1984 [14], *multivariable optimal constrained control algorithm* (MOCCA) due to Sripada and Fisher in 1985 [15] and *generalized predictive control* (GPC) due to Clarke *et al.* in 1987 [16, 2].

The main motivation for the development of LRPC algorithms was due to the challenges posed by real processes to achieve proper control and operation. It is well known that most chemical processes are characterized either by time delays, non-linearities, inverse response or other unusual high frequency dynamics, unmeasurable disturbances, process constraints and interaction from other process variables or a combination of these. In general LRPCs can elegantly handle these difficult process situations unlike PID controllers. Although PID controllers are extensively used in the chemical industry for historical and simplicity reasons, they have many disadvantages that are widely acknowledged to be overbearing. For difficult plant situations, some disadvantages of PID controllers are: (a) poor performance (hence implications on operating cost) due to heavy detuning; (b) time consuming and nontrivial tuning processes; (c) inability to handle operating or instrument constraints; and (d) applicable only to SISO systems. It is not surprising that the initial LRPC algorithms such as IDCOM, DMC and MAC were developed by industrial practitioners and these controllers rapidly received wide-spread acceptance in the chemical industry [17, 18, 19].

Besides handling difficult plant situations, the LRPC algorithms can be easily extended to: (a) multivariable control strategies, (b) incorporate operating constraints and (c) integrate with other supervisory controllers that are based on economic objectives. For example, MOCCA is a multivariable controller and has an elegant way of handling process constraints. Other LRPC schemes such as DMC have been modified to LDMC[1] [20] and QDMC[2] [21] and similarly GPC has been modified to handle input and output constraints [22, 23]. It is interesting to note that Kishi as early as in 1964 considered constraints in model based control that are similar to QDMC or constrained GPC.

Different LRPC schemes differ in detail but they have the following common features: (a) irrespective of the model types (*i.e.* state-space, transfer function, step-response etc.), the anticipated or the predicted process output is in terms of the step or impulse response coefficients [24]; (b) the predicted process output comprises of a forced response and a free response term; (c) the control actions are calculated so as to minimize a user specified objective function; and (d) receding horizon strategy is used to implement the controller.

---

[1] LDMC is an abbreviation for Linear programming method of optimization with DMC [20].

[2] QDMC is an abbreviation for Quadratic programming solution of DMC [21].

1.2 Motivation and objective

The research in LRPC has attained a significant level of maturity. In many chemical plants, LRPCs such as DMC, IDCOM and GPC are operational[3]. But one issue of practical importance and interest that has not been actively pursued is the issue of robustness of LRPCs to modelling errors. This issue is important because, modelling errors manifest themselves as performance and stability problems in all certainty equivalence controllers such as LRPCs.

In a real situation a mathematical model can never emulate the true plant dynamics hence modelling errors or model-plant-mismatch (MPM) is inevitable. The reasons for the existence of the modelling errors are as follows:

- In real application, the plant often has infinite order, undefined structure and complex dynamics. Often it is difficult to describe such complex plant dynamics even by a first-principle based mechanistic models.
- Most LRPC algorithms are based on simple linear input-output models of finite dimension to obtain simple implementable controller design. As would be expected, for simple models, the extent of MPM is often significant.
- Most processes are corrupted by different types of unmeasured disturbances, which makes model identification difficult. In the presence of disturbances, the identified model is usually biased, which then gives rise to MPM.

The next issue of importance is how MPM should be characterized. The MPM can be characterized parametrically as well as nonparametrically. Parametric MPM could be mismatch in process gain, time delay or time constant. But often for complex processes it is not possible or feasible to specify parametric bounds on process dynamics. The non parametric uncertainty on the other hand lumps the mismatches in gain, delay, time constant and other unusual dynamics together. Such a non parametric uncertainty that encompasses all possible type of mismatch is conveniently represented in the frequency domain. Therefore one objective of this thesis becomes:

Objective - 1

To estimate non-parametric MPM in the frequency domain from time domain process data.

The problem of robustness of LRPC to MPM was recognized as early as 1964 by Kishi [3], but he did not specify any methods on how to handle this situation. Later in the context of MAC, Rouhani and Mehra suggested a time domain based robustness measure [9]; which can only be tested by performing closed-loop experiments. But here the question raised is: can the robustness of LRPC be ascertained without performing any closed loop experiments? In order to address this issue, another endeavor of this thesis becomes:

---

[3] The most successful LRPC controllers in the industry are DMC and IDCOM which are respectedly marketed by DMC Corporation and Setpoint Inc., (both these companies have now been purchased by ASPEN TECH) both located in Houston, USA. GPC is more popular in academia, nevertheless, some applications of GPC have been reported in the literature [16, 18].

Objective - 2

To design LRPC that is robust to the non-parametric MPM and ascertain the robustness of such a design without performing any closed loop tests.

Campo and Morari considered a family of process models and used them to formulate an optimal problem for the LRPC that minimizes the worst case tracking error [25]. The problem is solved using non-convex minimax optimization problem, for which efficient algorithms are not available [25]. Moreover they used time domain based uncertainties for which the robust design methods are not yet fully developed. *Instead of considering a family of models (which involves more work in system identification), the present work focuses on using a single model and its associated non-parametric MPM to design robust LRPC.*

A number of results and design techniques are available in the area of robust control design that address the problem of controller design that is robust to MPM [26, 27, 28]. One such simple tool is the *small gain theorem* (SGT) that can be used to assess the robustness of a model based controller in the presence of MPM and such a design is carried out in the frequency domain [26, 28]. *SGT is therefore used as a tool in this thesis to ascertain the stability of LRPC to the MPM and this analysis is carried out in the frequency domain.*

Robust design of linear quadratic (LQ) optimal controllers within the SGT framework is mentioned in Bitmead *et al.* (1990) [29]. But the use of the SGT to examine the robustness of LRPC was probably first proposed by Robinson and Clarke in the context of robust design of GPC [30]. They used the SGT criteria to establish robustness of one of the important tuning parameters of GPC. However for a robust LRPC design one should examine the robustness of all the tuning parameters; hence the thesis objective expands to:

Objective - 3

To examine the effect of all tuning parameters on the robustness of LRPC.

Another issue of importance that is associated with robust design is performance, because, a good performance is what ultimately matters. For certainty equivalence controllers, robust stability is obviously a necessary prerequisite for a good performance. In this context, several control-relevant identification methods have emerged in the early 1990s [31, 32, 33] and the research interest in this area is presently growing. In order to make this study complete, the following additional objective is also considered in this thesis:

Objective - 4

To examine different control relevant identification methods with a robustness perspective and evaluate them for use with LRPC methods.

The objectives discussed above define the main goals of this thesis. The results for this thesis are summarized in Chapter 8.

1.3 Scope

The study of robustness and performance of LRPCs is intimately connected with signal processing, system identification, robust design tools, LRPC design and control relevant identification methods. Therefore to carry out this study, it is required to define the boundaries of these related issues, which is done in the following:

Signal processing

The robust design method used in this work requires a description of the MPM in the frequency domain. Such a description of the MPM can be obtained using the signal processing methods. Both parametric as well as non parametric signal processing methods are available to characterize a signal or a system. Only non parametric signal processing methods such as the use of discrete Fourier transforms (DFT) are considered because they are simple to use and reliable. Moreover for noisy systems, parametric signal processing methods do not provide any particular advantage [34]. Both time and frequency domain signal processing methods are considered because: (a) time domain signal processing is useful for model validation; and (b) robust design methods require system descriptions in the frequency domain.

System identification

The design of any LRPC is based on a given model based description of the plant. Hence for an effective controller design, there is a need for a good quality model which can be obtained using system identification methods. The two main components of system identification are: (a) structure of the model considered; and (b) methods for estimating the model parameters. Discrete models considered in this thesis are transfer functions (ARIMAX, BJ etc), step response, orthonormal functions and state space models. Except for the state space model, other models are considered for identification studies. LRPC formulations discussed in this thesis are based on the transfer function and orthonormal function models. Orthonormal function models are converted into state space form to formulate LRPC. The identification methods considered are ordinary least squares, augmented UD method [35] and prediction error methods.

Robust design tool

As mentioned in the previous section, the small gain theorem is used to analyze the robustness of LRPC in the frequency domain.

LRPC design

The LRPC algorithms considered for robustness analysis in this thesis are: (a) transfer function (ARIMAX) model based GPC and (b) a DMC like controller which is based on an orthonormal fuction model (specifically Markov-Laguerre model). Most model predictive controllers have been shown in the literature to be specal cases of GPC [2, 16, 36]; hence GPC is considered in this thesis as the main representative algorithm for the LRPC. The DMC scheme is selected because: (a) it is widely used in the processs industries and (b) it is somewhat structurally different from GPC as it is based on an unstructured model. In order to show the similarities in the robustness properties for different LRPC structures, both GPC and DMC type controllers are considered in this study.

Control relevant identification

The three control relevant identification methods due to Shook and Shah, Rivera *et al.* and Zang *et al.* are considered [31, 32, 33] and these methods are evaluated by their application to GPC.

1.4 Structure

The thesis is divided into eight chapters including the introductory chapter. Chapters 2 to 7 are relatively independent from one another hence they can be read in any order according to the reader's preference. There are some cross dependencies between these chapters, but they are marginal and they can be traced with minimum effort. The contents of these chapters are summarized in the following:

Chapter 2: The idea of signal processing is presented with a historical perspective. The time and frequency domain aspects of signal processing are tutorially introduced and they are combined with the statistical methods to verify the estimates. The use of signal processing methods to estimate the spectrum of MPM from open loop time domain data are also discussed in this chapter.

Chapter 3: Various system identification methods are reviewed to estimate different process models. Model estimation and validation techniques are illustrated by their application to two industrial case studies.

Chapter 4: Orthonormal function models are introduced with a historical perspective. The augmented UD identification method is used in this chapter to simultaneously estimate parameters of different orders of orthonormal function models in one computational step. This method is applied to different models such as Laguerre, Kautz, FIR and Markov-Laguerre models.

Chapter 5: The use of the small gain theorem to obtain robust design guidelines for GPC is discussed in this chapter. Through simulations, analytical methods and experimental evaluations it is shown how different tuning parameters affect the robustness of GPC. A method is also proposed to estimate MPM and the process model from GPC closed loop data. It is also shown how optimization techniques can be combined with the SGT to automatically select robust tuning parameters.

Chapter 6: A DMC like LRPC is formulated in this chapter by converting an estimated Markov-Laguerre model into a state space form. The concept of steady state weighting [37] is used to enhance the stability of this controller. This chapter also shows how a structured noise model can be combined with the unstructured Markov-Laguerre model to obtain an LRPC that gives faster disturbance rejection. SGT is used to illustrate that the robustness properties of this controller and GPC are similar.

Chapter 7: Control relevant identification methods due to Shook and Shah, Rivera *et al.* and Zang *et al.* [31, 32, 33] are reviewed in this chapter. Zang *et al.*'s concept of iterative design is extended to Shook and Shah and Rivera *et al.*'s identification strategies. A modified version of GPC is presented in this chapter that can be used to iteratively improve the controller performance. These methods are evaluated by their application to GPC.

Chapter 8: This chapter summarizes the conclusions presented in Chapters 2 to 7 and discusses future directions in research.

Bibliography

[1] J. Dawkins and P.A.N. Briggs. "A method for using weighting functions as system description in optimal control". In *Proceedings of IFAC Symposium*, Teddington, UK., 1965.

[2] D.W. Clarke and C. Mohtadi. "Properties of generalized predictive control". *Automatica*, 25:859–875, 1989.

[3] F.H. Kishi. *On line Computer Control Techniques and their Application to reentry Aerospace Vehicle Control.* Academic Press, New York, USA, 1964. Advances in Control Systems, Theory and Applications, Pt. 1., Series Ed: C.T. Leondes.

[4] R.E. Kalman. "Design of self-optimizing control system". *Trans. ASME*, 80:468 – 478, 1958.

[5] C.W. Merrium. "Use of a mathematical error criterion in the design of adaptive control systems". *Trans. AIEE Pt II*, 79:506 – 512, 1960.

[6] S. Horing. "On the optimum design of predictive control systems". In *1962 WESCON, August*, Los Angeles, CA, USA, 1962.

[7] J. Richalet, A. Rault, J.L. Testud, and J. Papan. "Model predictive heuristic control: applications to industrial processes". *Automatica*, 14:413 – 428, 1978.

[8] C.R. Cutler and B.L. Ramaker. "Dynamic matrix control-a computer control algorithm",Paper No. WP5-B. In *Proceedings of Joint Automatic Control Conference*, San Francisco, CA, U.S.A., 1980. Also published in 86th AIChE National Meeting, Paper No. 51B,Houston, USA, 1979.

[9] R. Rouhani and R.K. Mehra. "Model algorithmic control (MAC)". *Automatica*, 18:401 – 414, 1982.

[10] V. Peterka. "Predictor based self tuning control". In *Proceedings of* 6^{th} *IFAC Symposium on Identification and System Parameter Estimation*, 1982.

[11] V. Peterka. "Predictor based self tuning control". *Automatica*, 20:39 – 50, 1984.

[12] B.E. Ydstie. "Extended horizon adaptive control". In *Proceedings of* 9^{th} *IFAC World Congress*, Budapest, Hungary., 1984.

[13] J.M. Lemos and E. Mosca. "A multipredictor based LQ self-tuning controller". In *Proceedings of* 7^{th} *IFAC Symposium on Identification and System Parameter Estimation*, 1985.

[14] C. etal. Greco. "Performance improvement of self-tuning controllers by multistep horizons: The MUSMAR approach". *Automatica*, 20:681 – 699, 1984.

[15] N.R. Sripada and D.G. Fisher. "Multivariable Optimal Constrained Control Algorithm (MOCCA): Part 1. Formulation and Application". In *Proceedings of International Conference on Industrial Process Modeling and Control*, volume 1, Hangzhou, China., 1985.

[16] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. "Generalized Predictive Control-Part I and II". *Automatica*, 23:137–160, 1987.

[17] C.R. Cutler and R.B. Hawkins. "Application of large predictive controller to a hydrocracker second stage reactor". In *Proceedings of American Control Conf.*, pages 284–291, Atlanta, USA., 1988.

[18] D.W. Clarke. "Application of generalized predictive control to industrial processes". *IEEE Control Syst. Mag.*, 8:49–55, 1988.

[19] C.E. Garcia, D.M. Prett, and M. Morari. "Model Predictive Control: Theory and practice - a survey ". *Automatica*, 25:335–348, 1989.

[20] A.M. Morshedi, C.R. Cutler, and T.A. Skrovanek. "Optimal solution of Dynamic matrix control with Linear Programming Techniques (LDMC)". In *Proceedings of Automatic Control Conference, June 19-21,*, Boston, MA, U.S.A., 1985.

[21] C.E. Garcia and A.M. Morshedi. "Quadratic Programming Solution of Dynamic Matrix Control (QDMC)". *Chemical Engineering Communications*, 46:73 – 87, 1986.

[22] T.T.C. Tsang and D.W. Clarke. "Generalized Predictive Control with Input Constraints". *IEE Proceedings, Pt. D*, 135(6):451 – 460, 1988.

[23] R.K. Mutha. *Constrained Long Range Predictive Control.* MSc. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1990.

[24] S. Li, K.Y. Lim, and D.G. Fisher. "A State Space Formulation for Model Predictive Control". *AIChE Journal*, 35(2):241 – 249, 1989.

[25] P.J. Campo and M. Morari. "Robust Model Predictive Control". In *Proceedings of Automatic Control Conference, June 10-12*, Minneapolis, U.S.A., 1987.

[26] G. Zames. "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems. Part-I: Conditions Using Concepts of Loop Gain, Conicity and Positivity". *IEEE Trans. on AC*, 11(2):228–238, 1966.

[27] C.A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties.* Academic Press, New York, USA, 1975.

[28] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. *Feedback Control Theory.* Macmillan Publishing Co., New York, USA, 1992.

[29] R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control - The Thinking Man's GPC.* Prentice Hall International Series., 1990.

[30] B.D. Robinson and D.W. Clarke. "Robustness effects of a pre filter in GPC". *IEE Proceedings-D*, 138:2 – 8, 1991.

[31] D.S. Shook, C. Mohtadi, and S.L. Shah. "A Control-Relevant Identification Strategy for GPC". *IEEE Trans. on AC*, 37(7):975 – 980, 1992.

[32] D.E. Rivera, J.F. Pollard, L.E. Sterman, and C.E. Garcia. "An Industrial Perspective on Control Relevant Identification". In *American Control Conference*, pages 2406–2411, San Diego, California, USA, 1990.

[33] Z. Zang, R.R. Bitmead, and M. Gevers. "H_2 Iterative Model Refinement and Control Robustness Enhancement". In *Proceedings of the* 30^{th} *CDC Meeting, December 1991*, Brighton, England, 1991.

[34] Kay. S.M. and S.L. Marple. "Spectrum Analysis - A Modern Perspective". *Proceedings of the IEEE*, 69(11):1380 – 1418, 1981.

[35] S. Niu, D.G. Fisher, and D. Xiao. "An Augmented UD Identification Algorithm". *Int. J. Control*, 56(1):193 – 211, 1991.

[36] K. Kramer and H. Unbehaun. "Predictive Adaptive Control - Comparison of Main Algorithms". In *European Control Conference, July 2-5*, Grenoble, France, 1991.

[37] K.Y. Kwok and S.L. Shah. "Long-Range Predictive Control with a Terminal Matching Condition". *Chem. Eng. Sci.*, 49(9):1287 – 1300, 1994.

Chapter 2

Signal Processing for System Identification

2.1 Introduction

The example of splitting a sunbeam with a prisim into a rainbow of colored rays is often quoted to explain the intent of signal processing. In simple words, signal processing can be said to be the process of expressing a signal by sequences of numbers or symbols with a view to either: (a) estimate characteristic parameters of the original signal; or (b) transform the signal into a form which is more desirable [1].

An excellent example of a signal processor that nature has bestowed upon us is the ear. The ear transforms sound waves into signals of varying pitches for the brain to turn this information into perceived sound [2]. Signal processing as a mathematical tool has been applied to diverse areas such as seismology, astronomy, oceanography, acoustics, sonar, radar, quantum mechanics, telecommunication, medical science (*e.g.* cardiograms), nuclear science, meteorology, semiconductor physics, integrated circuit technology, opto-electronics, spectrometer for chemical analysis, speech processing, image processing, solution of partial differential equations, stock market data and many others [2, 3, 4, 5].

Signal processing plays a crucial role in many of these applications. For example, signal processing is extensively used to extract information from seismic data. Images beamed from satellites are enhanced using signal processing techniques. In fact signal processing has played a key role in laying the foundation of modern quantum mechanics.

It is well known that frequency response of a linear dynamic system is based on the principles of signal processing. It is needless to say that frequency response behaviour plays a crucial role in the design of classical linear controllers. Signal processing is also useful in system identification for validating the quality of estimated mathematical models. It is known that the quality of estimated models have a profound effect on the stability and performance of model based controllers [6]. Performance assessment of controllers is yet another emerging area of research where signal processing have proved to be useful [7].

---

[1] Versions of some sections of this chapter have appeared in: (a) P.Banerjee and S.L.Shah,'Estimation of Model-Plant uncertainty and its role in the Robust design of Predictive Control', *12th IFAC World Congress*, Sydney, Australia, vol-2, pp 321-326, July 18-23, 1993 and (b) P.Banerjee and S.L.Shah, 'The Role of Signal Processing Methods in the Robust Design of Predictive Control', *Automatica*, vol 31, No. 5, pp. 681-695, 1995.

The link between signal processing, data analysis and controller design have often been used to control electrical or mechanical systems. However in the chemical industry this potential of signal processing has not been actively pursued or applied to improve the control and performance of their process plants. Interestingly, petroleum industries have extensively relied on signal processing methods in their exploration for oil. In fact the major impetus in the development of modern signal processing can be attributed to those projects that eventually led to the discovery of major crude oil fields around the world [3]. The petroleum industries thus have the potential to derive further benefits from signal processing by using them to improve their plant operation via improved controller design and performance assesment.

The field of signal processing has a large number of well established results widely available in the engineering literature. These results are being used regularly for many applications described at the beginning of this section. The objective of this chapter is to consolidate in a tutorial manner some of the classical signal processing results that are of relevance to the field of system identification and control. This chapter has a tutorial flavor to motivate the chemical engineer reader to adopt signal processing methods so that their use will (hopefully) lead to better data analysis, a good process model, robust control design etc.

Most results presented in this chapter are complimented by illustrations to conform to the classical proverb: *pictures are worth thousand words.* The derivation of the most basic signal processing results are largely omitted because: (a) they are already well established in the literature; (b) the main interest lies in the application of these results; and (c) some of these derivations are very involved and do not fall within the scope of this thesis. However, a few useful and relevant results, a few key steps and the assumptions made in their derivation are presented to facilitate their significance.

Advances in signal processing has treaded through an interesting path. A brief historical sketch of these key developments in signal processing is included in the ensuing Section 2.2. A review of the historical development of signal processing is followed by a short introductory discussion on signal characteristics in Section 2.3. The time and frequency domain aspects of signals are then discussed in Sections 2.4 and 2.5 respectively. A separate Section 2.6 is devoted to statistical methods because they are used to put confidence bounds on the estimated parameters. Results presented in Sections 2.4, 2.5 and 2.6 are illustrated by case studies in Section 2.7. Section 2.8 discusses the use of signal processing methods to estimate modelling errors; which is followed by the concluding remarks in Section 2.9.

2.2 Signal processing - a historical account[2]

In ancient times signal processing concepts were used by the Babylonians to predict astronomical events and in 600 BC Pythagoras used them to understand the laws of musical harmony. In 1664 Sir Issac Newton empirically used signal processing concepts to discover the light spectra (published in 1704). German chemist Robert Wilhelm Bunsen (1811-1899) and subsequently Gustav Robert Kirchoff (1824-1887) extended Newton's experiment to discover the spectral D lines of the sodium ion, thereby laying the foundation for modern spectroscopy.

Pythagoras's interest with vibrating strings surfaced as a wave equation in the 18th century which set the tone for analytical developments in signal processing. L. Euler in 1748 noted that vibration of a string could be expressed as a linear combination of sinusoidal functions which was supported via analytical reasoning by D.Bernoulli in 1753. However J.L.Lagrange in 1759 heatedly contested this theory because he believed that a vibrating string being a nonanalytic function (*i.e.* a discontinuous function as both ends of string are fixed), it cannot be expressed as a combination of analytic functions such as the trigonometric functions.

Fourier Series

In the midst of this controversy, Baron Jean Baptist Joseph de **Fourier** (1768-1830), an engineer and a diplomat for Napolean Bonaparte, entered this fray that for all times marked the turning point in the development of signal processing. In an effort to solve the wave equation that characterizes the transient conductive heat behavior in an iron ring, Joseph Fourier concluded that any function (analytic or non-analytic) can be expressed as a *series of sinusoidal functions.* Fourier presented his results in 1807, but this result was (expectedly!) challenged by Lagrange and was not allowed to be published[3]. Fourier's work eventually got publicized when his book "The Analytical Theory of Heat" was released in 1822.

Fourier's mathematical reasoning about his *series* was imprecise although he had a clear insight into the problem. Later S.D.Poisson and A.L.Cauchy extended Fourier's work to prove convergence. It was P.L.Dirichlet in 1829 who finally provided a formal mathematical framework to the Fourier series which is now known by the Dirichlet's condition of Fourier series. Dirichlet's work provided the recognition to Fourier but this did not occur until near the end of Fourier's life.

---

[2] The Section *'Signal processing - a historical account'* is largely a synthesis of the material published in [2, 3, 4, 5, 8]. Therefore no specific references are cited in this section. The interested reader is encouraged to consult these references for further details.

[3] To be precise, Fourier presented his results on 21 December, 1807 at the prestigious French Academy of Sciences meeting where Lagrange stood up to challenge Fourier's result. A committee of four that included P.J.Laplace and J.L.Lagrange refused to publish Fourier's work after examining it; mainly due to Lagrange's opposition although Laplace and others favoured publication. Fourier's work was published 15 years after he first presented his result at the French Academy.

Amongst the initial users of Fourier's work into their fields were Claude Navier and Lord Kelvin. Lord Kelvin used Fourier series to predict tidal waves. G.G.Stokes in 1879 extended the concept of Fourier series to the *Fourier Transform* to find hidden periodicities in non-periodic signals. In 1894 A.Schuster proposed to consider modulus square of finite Fourier transform to detect hidden periodicities in a signal and he defined this term as a *periodogram*.

Fourier's work played an important role in solving primarily: (a) periodic solutions of partial differential equations (*e.g.* wave equation); (b) converting differential equations into algebraic equations using Fourier transform; and (c) approximating non-periodic functions by a set of orthogonal functions. (*Note: This thesis is mostly concerned with the third application. In particular the focus of this chapter is on representing signals by a set of sinusoidal functions.*)

Later developments

A landmark was reached when French mathematicians Charles Strum (1803-1855) and Joseph Louville (1809-1882) linked the Fourier series to a set of othonormal functions in the spectral theory of differential equations[4]. Lord Rayleigh (1842-1919) used Strum-Louville results to study elastic vibrations of solids and Erwin Schrödinger (1887-1961), in 1926, used the same results to explain vibrations within an atom and thereby accounting for the atomic spectral lines as experimentally observed by Bunsen. Werner Heisenberg (1901-1976) also arrived at Schrödinger's result a year earlier through a different route. Thus Heisenberg's and Schrödinger's results laid the foundation for modern quantum mechanics[5]. Later John von Neumann[6] (1903-1957) who introduced the abstract Hilbert space[7] in quantum mechanics (in 1929), mathematically showed the indispensability of spectral analysis in quantum mechanics. Fourier analysis was also used to obtain finer resolution of the atomic spectral lines, that were not noticed by Bunsen.

Norbert Wiener used Fourier analysis to mathematically describe Brownian motion

---

[4] Chapters 3 and 4 consider orthonormal functions in more detail.

[5] In simple words, according to quantum mechanics: (a) changes in energy level of electrons manifest as electomagnetic energy which can be quantified by photons and shows as spectral line at some frequency; (b) electrons in an atom are given by probability density function and their shape determines the probability of electron jump; and (c) the sharpness of spectral lines are explained using Heisenberg's uncertainty principle and wave property of electron. Further discussion on this subject is beyond the scope of this thesis.

[6] Neumann was a mathematician who worked in Physics.

[7] Hilbert space is an infinite dimensional space which is characterized by an inner product. Hilbert space is used to represent orthonormal function as discussed in Chapter 4.

(first reported by Robert Brown in 1827) which is now known as the Einstein-Weiner theory and it forms the basis for the white noise signal. The use of Fourier analysis reached another milestone in 1962, when it was used along with X-ray diffraction to discover the structure of DNA.

Digital signal processing

An early version of a numerical technique to compute the nonparametric spectrum of a time series data was introduced by Sir Arthur Schuster in 1898. Schuster's method did not work for random signals, so G. Udny Yule in 1927 proposed a parametric model which he called the ***autoregressive model*** to represent the signal spectrum.

Much before the arrival of powerful computers, N.Levinson in the 1940's converted Wiener's work on continuous signal processing into discrete domain. The advent of digital computers in the mid 1950s revolutionized the use of spectral analysis. Digital signal processing was first applied to Mobil Oil's seismic data records by the MIT Geophysical Analysis Group where J.W. Tukey contributed significantly towards the development of numerical methods to compute the discrete Fourier transform. The use of digital signal processing of seismic data records led to the discovery of major oil fields around the world.

The development of the Fast Fourier Transform (FFT) algorithm by Cooley and Tukey in 1965 was a hallmark in the development of the discrete Fourier analysis. Since then several forms of FFT algorithms have appeared. The FFT algorithms reduced the computational burden significantly, so consequently it helped in spreading the use of signal processing to diverse areas. Application of FFT methods have now become very trivial because software for the FFT algorithms are widely available in the market for almost all type of computer platforms. Further, even hardware implementation of FFT are now available that can process a huge amount of data at a much faster rate than the FFT software.

2.3 Signal characterization

The objective of this section is to give a brief introduction to the problem of signal processing and the nomenclature for signal characterization. Both continuous and sampled signals are analyzed using signal processing, but the emphasis here is on discrete signal processing because the focus of this thesis is on the design of digital controllers.

The reference system block diagram around which the signal processing methods are applied in this thesis is shown in Figure 2.1. The blocks $G(z)$ and $\hat{G}(z)$ in Figure 2.1 denote a *discrete linear time invariant* plant and its model respectively. A description of various signals in this block diagram are listed in the Section *Notations* but in addition the following notes also apply:

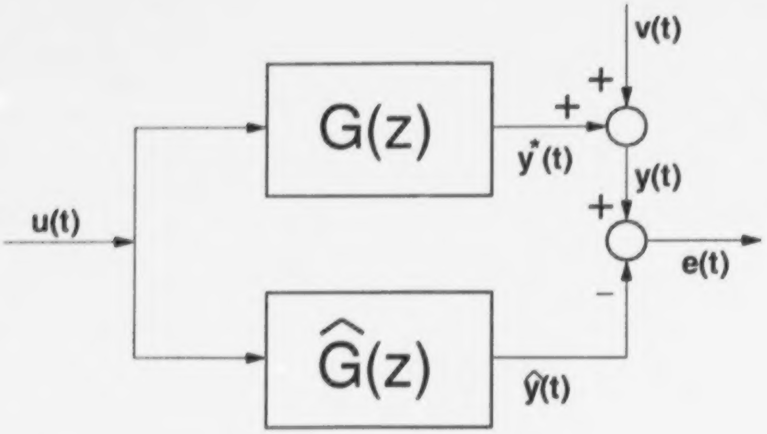


Figure 2.1: Block diagram of plant and model.

Note 2.1

- *x generically denotes any signal in Figure 2.1. This means that if a property holds for x, then that property will also hold for other signals in Figure 2.1.*
- *$x(t, N)$ denotes a discrete time series at points $t = 1 \ldots N$. Continuous time data are represented by $x_c(t, T)$ where T is the time period of the signal. If $x(t, N)$ and $x_c(t, T)$ are of same duration then $T = NT_s$ where T_s is the sampling time.*

The signal $x(t, N)$ or $x_c(t, T)$ can either be ***stochastic*** or *deterministic*. In case of a stochastic signal, $x(t, N)$ is independent of t whereas $x(t, N)$ is a function of t for a deterministic signal. Mathematical models apply to deterministic signals whereas stochastic signals are analyzed using statistical methods. Stochastic processes are assumed to be: (a) ***stationary***; (b) obey certain ***probability distribution***; and (c) are characterized by some ***mean*** and ***variance***. A time series which is in statistical equilibrium and devoid of any ***trends*** is called a ***stationary series***. It is assumed that a stationary stochastic process is ***normally distributed*** and it is designated by $\mathcal{N}(\mu, \sigma^2)$ where μ and σ^2 are the mean and variance of the time-series respectively.

For an *openloop* system of the type shown in Figure 2.1, $u(t, N)$ can be either stochastic or deterministic. But if the filter or transfer function $G(z)$ exists, then it will make the output $y^*(t, N)$ a deterministic signal. Similarly, the noise or disturbance sequence $v(t, N)$ can either be stochastic or deterministic; but in a real situation $v(t, N)$ is usually unknown. Further, $v(t, N)$ is assumed to corrupt $y^*(t, N)$ such that the *observable* or measurable plant output is $y(t, N) = y^*(t, N) + v(t, N)$. Therefore one of the tasks of the signal processing technique is to extract the information, $y^*(t,N)$, from the signal $y(t,N)$. The system in Figure 2.1 can be represented as:

$$\begin{aligned} y(t) &= \underbrace{G(q^{-1})u(t)}_{y^*(t)} + v(t) \\ \hat{y}(t) &= \hat{G}(q^{-1})u(t) \end{aligned} \tag{2.1}$$

The following remarks apply to the above equation-(2.1) and Figure 2.1.

Remark 2.1

- *The argument z in $G(z)$ denotes frequency domain i.e. $z = e^{-j\omega T_s}$.*
- *The argument q^{-1} in $G(q^{-1})$ is called the* back-shift *operator and similarly q denotes the* forward-shift *operator (see the Notation section). The operators q or q^{-1} are used for time domain manipulations.*
- *This thesis uses q^{-1} operator because it is standard for system identification.*
- *z is analogous to s in the Laplace domain, whereas q^{-1} is analogous to the differential operator d/dt in the continuous time domain. For the discrete case, z and q^{-1} are interchangable because the differential equation continuous in the domain simplifies to a difference equation in the z domain.*

Stochastic or deterministic signals can be analyzed in time and/or frequency domains. The following sections show how time and frequency domain signal characterizations are interrelated.

2.4 Signal processing in time domain

A measure of dependency of a signal at any two sampling instants is given by the *autocovariance* or by its normalized version *i.e.* *autocorrelation* function. Similarly correlation between two signals is given by *cross covariance* or its normalized version *i.e.* *cross correlation* function. The concept of cross correlation can be extended to any number of signals where it becomes the subject of multivariate statistical analysis. These covariance/correlation terms are defined in the following subsections 2.4.1 and 2.4.2.

---

[8] Key references for the Section *'Signal processing in time domain'* are [9, 10].

2.4.1 Autocovariance/Autocorrelation

Autocovariance function of a signal $x(t, N)$ is given by [10]

$$\gamma_{xx}(k) = \mathcal{E}[(x(t) - \mu)(x(t+k) - \mu)] = \mathrm{cov}[(x(t) - \mu), (x(t+k) - \mu)] \tag{2.2}$$

where μ is the mean *i.e.* $\mu = \mathcal{E}[x(t, N \to \infty)]$. In practice N is finite, therefore a *sample mean* of $x(t, N)$ *i.e.* $\bar{x} = \frac{1}{N}\sum_{t=1}^{N} x(t)$ is generally used. For a finite data record, the sample autocovariance function [10] is given by:

$$\hat{\gamma}_{xx}(k) = \frac{1}{N}\sum_{t=1}^{N-k}(x(t) - \bar{x})(x(t+k) - \bar{x}) \tag{2.3}$$

where ˆ in $\hat{\gamma}_{xx}(k)$ denotes a sample or approximation of the true $\gamma_{xx}(k)$. In order to compare different time series and make this term measure unit independent it is often convenient to normalize the autocovariance function by dividing it by variance *i.e.* $\mathrm{var}(x) = \gamma_{xx}(0) = \mathcal{E}[(x-\mu)^2]$. The normalized autocovariance is called *autocorrelation* function or simply *acf* in abbreviated form. The acf is defined [9] as:

$$\rho_{xx}(k) = \frac{\gamma_{xx}(k)}{\gamma_{xx}(0)} \tag{2.4}$$

Similarly the sample acf of $x(t, N)$ is [9]:

$$\hat{\rho}_{xx}(k) = \frac{\sum_{t=1}^{N-k}(x(t) - \bar{x})(x(t+k) - \bar{x})}{\sum_{t=1}^{N}(x(t) - \bar{x})^2}, \qquad k = 0, 1, 2, \ldots \tag{2.5}$$

2.4.2 Cross covariance/cross correlation

The cross covariance function between two signals say $u(t, N)$ and $y(t, N)$ can similarly be defined [10] as:

$$\gamma_{yu}(k) = \mathcal{E}[(y(t) - \mu_y)(u(t+k) - \mu_u)] = \mathrm{cov}[(y(t) - \mu_y)(u(t+k) - \mu_u)] \tag{2.6}$$

where $\mu_y = \mathcal{E}[y(t, N \to \infty)]$ and $\mu_u = \mathcal{E}[u(t, N \to \infty)]$. As in case of acf, the normalized cross covariance *i.e.* cross correlation function (or ccf) is given by:

$$\rho_{yu}(k) = \frac{\gamma_{yu}(k)}{\sqrt{\gamma_{yy}(0)\gamma_{uu}(0)}} = \mathrm{corr}[(y(t) - \mu_y)(u(t+k) - \mu_u)] \tag{2.7}$$

Based on equations-(2.3), (2.5), (2.6) and (2.7), the *sample ccf* becomes:

$$\hat{\rho}_{yu}(k) = \frac{\sum_{t=1}^{N-k}(y(t) - \bar{y})(u(t+k) - \bar{u})}{\sqrt{\sum_{t=1}^{N}(y(t) - \bar{y})^2 \sum_{t=1}^{N}(u(t) - \bar{u})^2}}, \qquad k = \ldots, -2, -1, 0, 1, 2, \ldots \tag{2.8}$$

The properties of acf and ccf are quite different [10]; some of them are:

- The acf is an even function of the lag k *i.e.* it is symmetric about $\rho_{xx}(0)$ or in other words $\boxed{\rho_{xx}(k) = \rho_{xx}(-k)}$. The ccf on the other hand is an odd function *i.e.* the ccf is non-symmetric around $\rho_{yu}(0)$ and $\boxed{\rho_{yu}(k) = \rho_{uy}(-k)}$.
- Maximum value of acf is 1 which is at lag 0. In contrast, the relation $|\rho_{yu}(k)| \leq 1$ holds good for ccf and its maximum can occur at any lag k.
- The acf helps to find the energy distribution and/or randomness of a signal whereas the ccf determines the dependency between two signals. The dependence between two signals is given by *coherence* and *phase*. The concepts of energy distribution, randomness, coherency and phase are discussed in the following sections.

2.4.3 Applications of acf

Some applications of acf that are useful for system identification are highlighted in this subsection.

Randomness or whiteness of a signal

If $x(t,N) \in \mathcal{N}(0,\sigma^2)$, then that implies $\rho_{xx}(k) = 0$ for $k \geq 1$. However for a finite N often $\hat{\rho}_{xx}(k)$ reduces to small values for $k \geq 1$ rather than to zero because $\hat{\rho}_{xx}(k)$ is only an approximation of $\rho_{xx}(k)$. Hence the whiteness test for finite N is always associated with some statistical confidence bounds and for $x(t,N) \in \mathcal{N}(0,\sigma^2)$, the bounds [9, 10]

$$\bar{\underline{\rho}} = (\rho_{xx}^U, \rho_{xx}^L) = \pm\frac{1.96}{\sqrt{N}} \tag{2.9}$$

indicate a confidence level of 95%, *i.e.* if $\hat{\rho}_{xx}(k) \in \bar{\underline{\rho}}$ then with a 95% confidence it can be said that the data are normally distributed. The bounds in equation-(2.9) is based on the assumption that $\rho_{xx}(k) - \hat{\rho}_{xx}(k) \in \mathcal{N}(0,\sigma^2)$. $\bar{\underline{\rho}}$ in equation-(2.9) is displayed by a pair of dashed lines in the acf plot in Figure 2.2. Figure 2.2 shows: (a) a portion of white noise sequence (given signal length is $N = 1024$); and (b) $\hat{\rho}_{xx}(k) \in \bar{\underline{\rho}}\ \forall\ k \geq 1$, thus confirming with a 95% certainty that the given time domain data are normally distributed. *This result is useful in validating process models as will be seen in Chapter 3.*

Moving Average (MA) signals

A moving average (MA) signal is related as [9]:

$$x(t) = \xi(t) - c_1\xi(t-1) - c_2\xi(t-2) - \cdots \tag{2.10}$$

where $\xi(t,N) \in \mathcal{N}(0,\sigma^2)$ and $c_1, c_2, \ldots$ are the MA parameters. If the order of MA process is r, then acf for such a MA(r) signal is given by [9]:

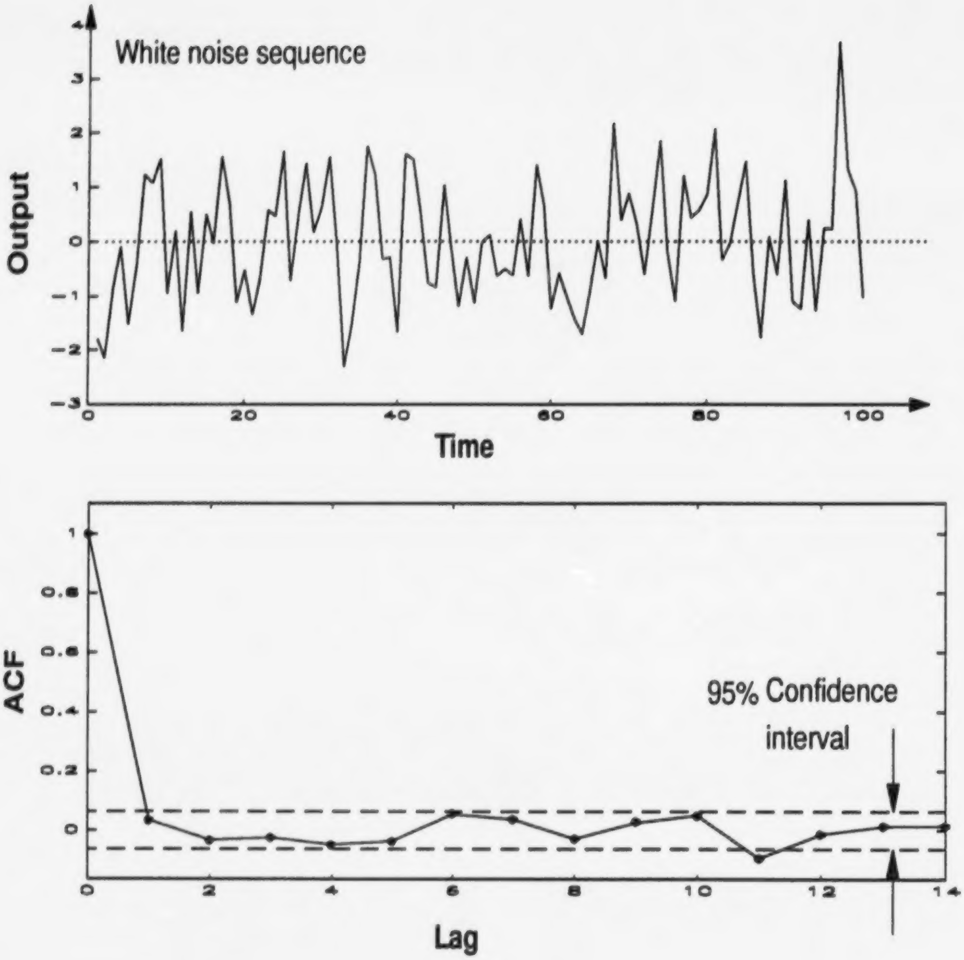


Figure 2.2: Top: White noise signal. Bottom: Autocorrelation of white noise signal.

$$\rho_{xx}(k) = \begin{cases} \frac{-c_k + c_1 c_{k+1} + c_2 c_{k+2} + \cdots + c_{r-k} c_r}{1 + c_1^2 + c_2^2 + \cdots + c_r^2} & \text{for } k = 1, 2, \ldots, r \\ 0 & \text{for } k \geq r + 1 \end{cases} \tag{2.11}$$

For a MA(r) process, equation (2.11) suggests: (a) $\hat{\rho}_{xx}(k) \in \bar{\underline{\rho}} \; \forall \; k > r$; and (b) acf can be calculated from the MA(r) parameters but it is difficult to determine MA(r) parameters from acf for $r > 1$. However MA(r) parameters can be inverted to AR process to indirectly obtain these MA(r) parameters [9]. Further discussion on the invertibility issue is omitted for brevity. The acf of MA(r) process is however useful in ascertaining its order r as illustrated in Figure 2.3.

Figure 2.3 shows a portion of MA(r) data generated by using: $v(t) = \xi(t) + 0.5\xi(t-1) + 0.3\xi(t-2)$ or compactly denoted by $v(t) = (1 + 0.5q^{-1} + 0.3q^{-2})\xi(t)$. The MA(2) data thus generated looks similar to $\xi(t)$ in Figure 2.2 but it shows more spread than $\xi(t)$ and nothing beyond. However an acf plot of this process in Figure 2.3 clearly reveals a MA(2) correlation because $\hat{\rho}_{xx}(k)$ is less than the 95% bound $\bar{\underline{\rho}}$ for all $k > 2$ or compactly this is denoted by $\hat{\rho}_{xx}(k) \leq \bar{\underline{\rho}} \; \forall \; k > 2$.

It is to be noted that the truncation in data manifests itself as an error between $\rho_{xx}(k)$ derived from equation-(2.11) (*i.e.* $\rho_{xx}(1) = 0.485, \rho_{xx}(2) = 0.224$) and their sample values (*i.e.* $\hat{\rho}_{xx}(1) = 0.497, \hat{\rho}_{xx}(2) = 0.182$ for $N = 1024$) determined by equation-(2.5) for the given MA(2) process. Further the bounds $\bar{\underline{\rho}} = \pm 1.96/\sqrt{1024}$ are applicable because of the assumption $\rho_{xx}(k) - \hat{\rho}_{xx}(k) \in \mathcal{N}(0, \sigma^2)$. *The procedure described in this section can be used to determine if the noise has a MA(r) structure.*

Autoregressive (AR) signals

An autoregressive signal of order p *i.e.* AR(p) is given by:

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots + a_p x(t-p) + \xi(t) \tag{2.12}$$

As an illustration, a time series of an AR(1) process generated using $(1 - 0.8q^{-1})v(t) = \xi(t)$ shows a much slower variation in Figure 2.4 than in Figures 2.3 or 2.2. The acf plot however provides a more clear picture where the acf function bears the following recursive relation [9]:

$$\rho_{xx}(k) = a_1 \rho_{xx}(k-1) + a_2 \rho_{xx}(k-2) + \cdots + a_p \rho_{xx}(k-p) \quad \forall k \geq 1 \tag{2.13}$$

A set of above equation-(2.13) for $k = 1, 2, \ldots, p$ is called the *Yule Walker* equations and unlike the MA(r) process, this equation can be used to determine the AR(p) parameters *directly* from the acf values. For example the sample acf values $[0.7909, 0.6046, 0.4649, \ldots]$

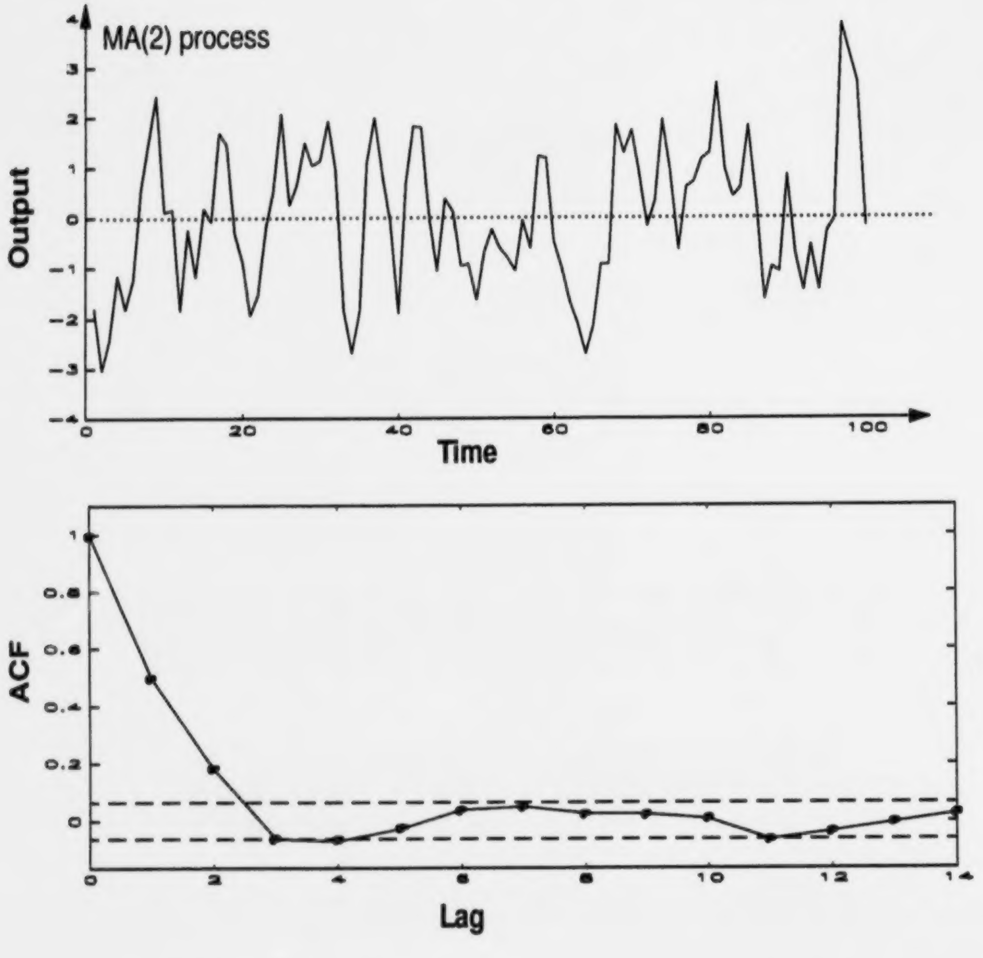


Figure 2.3: Top: MA(2) signal. Bottom: Autocorrelation of MA(2) signal.

(calculated using equation-(2.5)) in Figure 2.4 can be substituted in equation-(2.13) to obtain[9] $a_1 \approx 0.79$. The characteristics of AR(p) process is summarized in the following:

- The acf of an AR(p) process decays 'almost' exponentially. The decay in the acf can be either overdamped or underdamped depending upon the nature of the process.
- Equation-(2.13) can be used to determine the AR(p) parameters by using the acf values.
- The assumption $\rho_{xx}(k) - \hat{\rho}_{xx}(k) \in \mathcal{N}(0, \sigma^2)$ holds good for the AR(p) process.
- For an AR(p) process $\hat{\rho}_{xx}(k) \not\in \bar{\underline{\rho}}$ even for $k > p$ as illustrated in Figure 2.4 for $(1 - 0.8q^{-1})v(t) = \xi(t)$. Therefore for an AR(p) process, it is difficult to determine its order p from $\hat{\rho}_{xx}(k)$ but this difficulty is resolved by calculating the ***partial autocorrelation*** function (*pacf*), because the pacf of an AR(p) series behaves like an acf of MA(p) data.

For example the pacf plot in Figure 2.4 confirms that the given AR(p) process indeed has an order $p = 1$ and $a_1 \approx 0.79$. The pacf at lag k is calculated by removing the effect of all intervening lags [9, 10], *i.e.*:

$$\phi(k,k) = \mathrm{corr}(x(t), x(t-k) | x(t-1), x(t-2), \ldots, x(t-k+1)) \tag{2.14}$$

Levinson in 1947 showed that sample pacf can be recursively expressed [9] as:

$$\hat{\phi}(k,k) = \frac{\hat{\rho}_{xx}(k) - \sum_{i=1}^{k-1} \hat{\phi}(k-1,i)\hat{\rho}_{xx}(k-i)}{1 - \sum_{i=1}^{k-1} \hat{\phi}(k-1,i)\hat{\rho}_{xx}(i)} \tag{2.15}$$

where

$$\hat{\phi}(k,i) = \hat{\phi}(k-1,i) - \hat{\phi}(k,k)\phi(k-1,k-i) \quad \text{for } i = 1, 2, \ldots, k-1 \tag{2.16}$$

As a convention, the sample pacf recursion is initialized by setting $\hat{\phi}(1,1) = \hat{\rho}_{xx}(1)$. *As in case of MA(r) process, the acf or pacf can be used to ascertain if the noise has an AR(p) structure.* The following notes however apply:

Note 2.2

- *The acf or pacf plots are good only for the analysis of white noise, MA or AR processes.*
- *The acf/pacf plots are not adequate for the analysis of ARMA processes.*

---

[9] Note that: (a) equation-(2.13) reduces to $\rho_{xx}(k) = a_1^k$ for AR(1) process; and (b) at higher lags the accuracy of parameter estimation falls for AR(1).

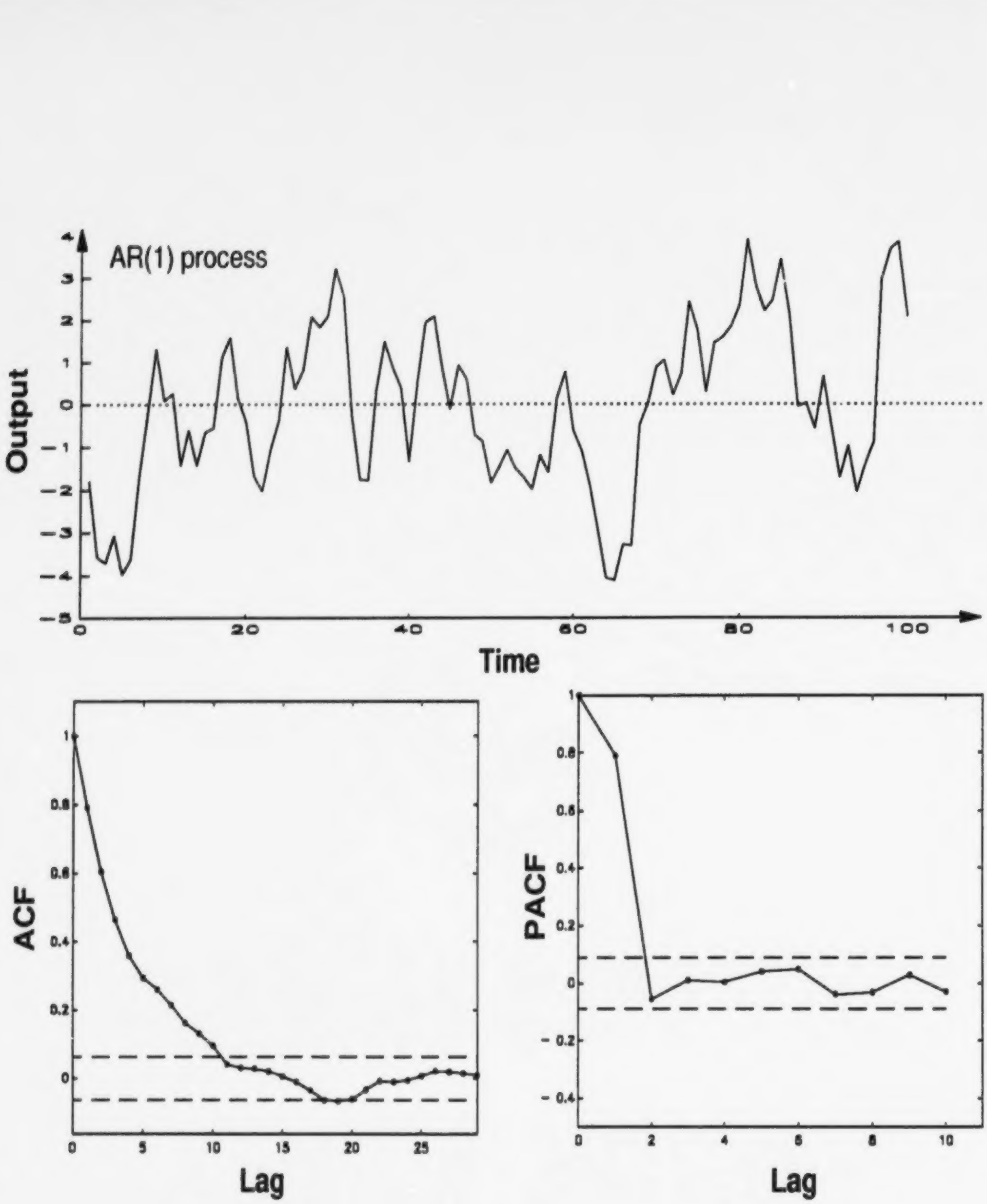


Figure 2.4: Top: AR(1) signal. Bottom-left: Autocorrelation of AR(1) signal. Bottom-right: Partial autocorrelation of the same signal.

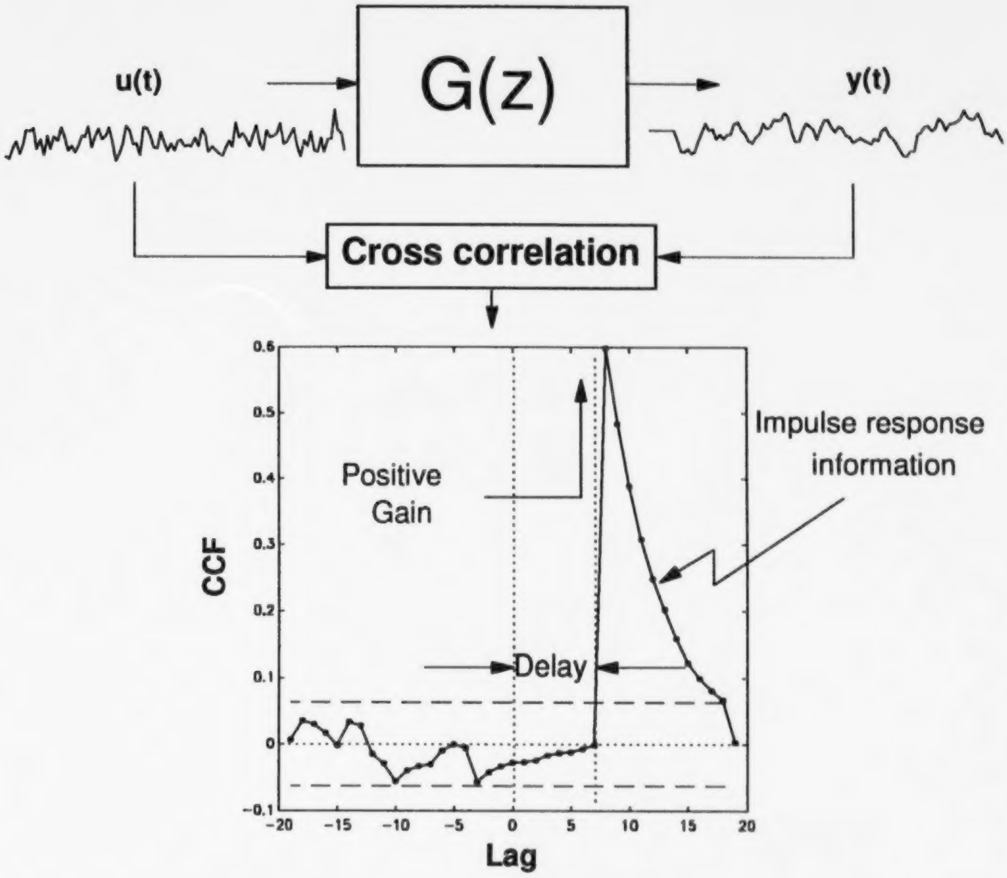


Figure 2.5: Application of cross correlation function to estimate the process characteristics.

- *For integrated: (a) white noise (i.e.* random walk *process); (b) MA; or (c) AR processes; the data must be differenced before doing the acf/pacf analysis. Similarly differentiation of appropriate order is required for higher order integrated processes.*

2.4.4 Applications of ccf

The ccf plots often provide useful information about the process transfer function and this is illustrated by considering the following plant:

$$G_1(q^{-1}) = \frac{0.2q^{-8}}{1-0.8q^{-1}} \tag{2.17}$$

Figure 2.5 shows a section of the system output $y(t, N = 1024)$ by subjecting the plant to a white noise excitation. The concept of confidence intervals $\bar{\rho}$ also applies here because the assumption $\rho_{yu}(k) - \hat{\rho}_{yu}(k) \in \mathcal{N}(0, \sigma^2)$ holds good. The cross correlation plot in Figure 2.5 conveys that:

- The ccf function is non symmetrical as discussed in subsection 2.4.2.
- At positive lags the ccf values are positive, thus indicating that the process gain is positive. Conversely, negative ccf values at positive lags would indicate a negative gain.
- $\hat{\rho}_{yu}(k) \in \bar{\underline{\rho}}$ for $k \in (0,7)$ and a peak value at lag 8 indicates that the process has a delay of 8 sampling intervals which is in conformity with equation-(2.17).
- The ccf values for lags beyond the process delay are related to the process impulse response in the following way:

  $$\mathbf{h} = \mathbf{A}^{-1}\mathbf{b} \tag{2.18}$$

  where $\mathbf{h}$ is a vector of impulse response coefficients, $\mathbf{A}$ is a symmetric matrix of $\hat{\rho}_{uu}(k)$s and $\mathbf{b}$ is a vector of $\hat{\rho}_{yu}(k)$s. The elements of $\mathbf{A}$ are arranged such that its first row is $\hat{\rho}_{yu}(0), \hat{\rho}_{yu}(1), \hat{\rho}_{yu}(2), \ldots$, second row is $\hat{\rho}_{yu}(1), \hat{\rho}_{yu}(0), \hat{\rho}_{yu}(1), \ldots$ and so on. In practice equation (2.18) is not used much because it gives significant variance in the estimated parameters. However a frequency domain equivalent of equation-(2.18) is widely used in practice as discussed in Section 2.5.

Further, if $\hat{\rho}_{yu}(0) \neq 0$ but $\hat{\rho}_{yu}(k) = 0$ for all other ks, then it can be inferred that the processes $y(t, N)$ and $u(t, N)$ are correlated at simultaneous times but otherwise uncorrelated.

2.5 Signal processing in frequency domain

In many applications including process control, much of the design analysis is carried out in the frequency domain. For example, robustness analysis of a controller, which is the theme of this thesis, is best done in the frequency domain. In Chapter 3 it is also shown that better mathematical (dynamic) models can be obtained by evaluating and validating them in the frequency domain. Similarly much of the work in filter design and image processing is done in the frequency domain. For such applications it becomes necessary to convert time domain data into frequency domain for the design or analysis to be possible.

The frequency domain representation is often more compact and reveals more information about the system than can possibly be obtained from time domain representation. The field of signal processing has tools that can be used to convert time domain signals into frequency domain in a wide variety of applications.

The objective of this section is to introduce *discrete Fourier transform* (DFT) which converts a *discrete signal* into *discrete frequency* domain. DFT in the form of FFT has become an indispensable tool in many applications, including this thesis. In order to understand DFT it is necessary to explain the concept of *continuous and discrete frequencies* and how they are related to *continuous and discrete signals*. These basic concepts are outlined in the ensuing subsections.

DFT represents the *classical* signal processing approach which directly provides nonparametric or unstructured frequency response from the time domain data. But there are also *modern spectral* methods which give parametric and structured models to represent the signal spectrum. Although modern spectral methods can be used to estimate the frequency behavior for a noise-free plant, these methods often yield no better results than the DFT method in the presence of noise [11]. Moreover the FFT approach is computationally quite efficient. In view of this, only the classical signal processing approach *i.e.* the DFT method has been addressed in this thesis.

2.5.1 Continuous time periodic signal

A signal in continuous time $x_c(t)$ is periodic with a periodicity T if $x_c(t) = x_c(t+T)$. Fourier series[10] can then be applied to such a periodic signal. Dirichlet showed that any periodic function $x_c(t,T)$ can be expressed as a Fourier series if $\int_T |x_c(t)|dt < 0$ *i.e.* the function is finite and the number of discontinuities in $x_c(t,T)$ are finite. This boundedness condition of Fourier series is called Dirichlet's condition and it is also applicable to a periodic discrete signal $x(t,N)$. Any periodic signal $x_c(t,T)$ obeying Dirichlet's condition can be expanded into the following Fourier series [12]:

$$x_c(t) = \frac{A_0}{2} + \sum_{k=1}^{\infty} A_k \cos(k\Delta\omega t) + \sum_{k=1}^{\infty} B_k \sin(k\Delta\omega t) \tag{2.19}$$

where $\Delta\omega = 2\pi/T$ is the fundamental frequency, A_0 is the mean average of $x_c(t)$ *i.e.* $A_0 = \frac{2}{T}\int_{-T/2}^{T/2} x_c(t)dt$ and A_k and B_k are the Fourier coefficients:

$$\begin{aligned} A_k &= \frac{2}{T}\int_{-T/2}^{T/2} x_c(t)\cos(k\Delta\omega t)dt \\ B_k &= \frac{2}{T}\int_{-T/2}^{T/2} x_c(t)\sin(k\Delta\omega t)dt \end{aligned} \tag{2.20}$$

For example a sequence of square waves with $T = 2\pi$ such that $x_c(t) = [1$ for $0 \leq t \leq \pi$ and 0 for $\pi \leq t \leq 2\pi]$ can be expanded into:

$$x_c(t) = \frac{1}{2} + \frac{2}{\pi}\left(\sin(t) + \frac{1}{3}\sin(3t) + \frac{1}{5}\sin(5t) + \cdots\right) \tag{2.21}$$

Equation-(2.19) (similarly equation-(2.20)) can be compactly expressed as [12]:

$$x_c(t) = \sum_{k=-\infty}^{\infty} X_c(k\Delta\omega)e^{jk\Delta\omega t} \tag{2.22}$$

where $j = \sqrt{-1}$ and $X_c(k\Delta\omega) = \frac{1}{T}\int_{-T/2}^{T/2} x_c(t)e^{-jk\Delta\omega t}dt$ is the frequency spectrum of the signal at discrete frequency points $k\Delta\omega$, for $-\infty \leq k \leq \infty$. The subscript c in X_c denotes it corresponds to the continuous signal $x_c(t)$. $X_c(k\Delta\omega)$ for $-\infty \leq k \leq \infty$ is a set of complex numbers *i.e.* $X_c(k\Delta\omega) = X_{cr}(k\Delta\omega) + jX_{ci}(k\Delta\omega)$ and it is also expressed as $|X_c(k\Delta\omega)|\angle X_c(k\Delta\omega)$ *i.e.* as a set of *magnitude* and *phase* spectrums. The important message of this subsection is that - ***even though the periodic signal is continuous, it's Fourier series is represented at discrete frequencies at intervals*** $k\Delta\omega$.

---

[10] as proposed by Joseph Fourier in 1807.

2.5.2 Non-periodic continuous signal

For a nonperiodic signal, it is assumed that its time period $T \to \infty$, as a result the fundamental (or incremental) frequency $\Delta\omega \to 0$ since $\Delta\omega = \frac{2\pi}{T}$. Therefore for a nonperiodic signal the effect of continuous frequency is realized and for such a case instead of Fourier series, Fourier transform (FT) applies [12], thus:

$$\begin{aligned} X_c(\omega) &= \int_{-\infty}^{\infty} x_c(t)e^{-j\omega t}dt \\ x_c(t) &= \int_{-\infty}^{\infty} X_c(\omega)e^{j\omega t}d\omega \end{aligned} \tag{2.23}$$

where $X_c(\omega)$ is the FT of $x_c(t)$. As in case of periodic signal, FT results in complex function $X_c(\omega) = X_{cr}(\omega) + jX_{ci}(\omega)$ for $-\infty \leq \omega \leq \infty$ and it may also be expressed as $|X_c(\omega)|\angle X_c(\omega)$. The Fourier transform is related to the Laplace transform because a simple substitution $s = j\omega$ translates Laplace transform into FT. But the Fourier transform in addition can also be applied to negative time, unlike the Laplace transform. An important conclusion of this subsection is that - ***FT converts nonperiodic continuous signal into a continuous function in the frequency domain.***

2.5.3 Non-periodic discrete signal

Discrete signals are obtained by sampling a continuous signal with a hypothetical train of unit pulse signals $\delta(t)$. Unit pulses in the train are separated by a sampling time T_s. Theoretically $\delta(t)$ has a unit area with effectively no width but in reality the sampler has a certain width τ. Let such a practical sampler be designated by $\hat{\delta}(t)$. Discrete signals in fact are obtained as $x(t) = x_c(t)\hat{\delta}(t)$. Further, the FT of $\hat{\delta}(t)$ is given by a *sinc* function *e.g.* $\text{sinc}(\omega\tau/2) = \tau\sin(\omega\tau/2)/(\omega\tau/2)$. Since τ has a narrow width, the sinc function drops off gradually in the frequency domain. Also $\hat{\delta}(t)$ being a periodic signal with a time period T_s, equation-(2.22) can be applied to express $\hat{\delta}(t)$ by the following Fourier series:

$$\hat{\delta}(t) = \underbrace{\sum_{-\infty}^{\infty} \frac{\tau}{T_s}\text{sinc}\left(\frac{k\omega_s\tau}{2}\right)}_{FT(\hat{\delta}(t))} e^{jk\omega_s t} \tag{2.24}$$

where the sampling frequency $\omega_s = 2\pi/T_s$. Since the discrete realization is $x(t) = x_c(t)\hat{\delta}(t)$, its frequency response is obtained by first multiplying equation-(2.24) by $x_c(t)$ as:

$$x(t) = \underbrace{\sum_{-\infty}^{\infty} \frac{\tau}{T_s}\text{sinc}\left(\frac{1}{2}k\omega_s\tau\right) e^{jk\omega_s t}}_{\hat{\delta}(t)} x_c(t) \tag{2.25}$$

and then taking its FT to finally yield:

$$X(\omega) = \sum_{-\infty}^{\infty} \frac{\tau}{T_s}\text{sinc}\left(\frac{1}{2}k\omega_s\tau\right) X_c(\omega - k\omega_s) \tag{2.26}$$

The above result is derived by using the property $FT(x(t)e^{j\omega_o t}) = X(\omega - \omega_o)$. Equation-(2.26) leads to the following important conclusions:

- Multiplication of two time series results in a convolution in frequency domain and *vice-versa*. For example compare equations-(2.25) and (2.26).
- *Even though the non-periodic signal is discrete, it translates into a continuous function in the frequency domain.*
- Spectrum of a discrete signal comprises of spectrum of the original continuous signal together with an infinite series of its clones.
- The cloned spectrums occur at multiples of sampling frequency ω_s.
- These discrete frequency spectrums are enveloped by the sinc function *i.e.* by $\frac{\tau}{T_s}\text{sinc}(k\omega_s\tau/2)$. This sinc function drops off as it moves away from $\omega = 0$, thus simultaneously reducing the magnitude of the cloned signal spectrums. For a very small sampling width τ, the drop off in the sinc function is very gradual, this results in a gradual decrease in the magnitude of the cloned spectrums and *vice versa.*
- As the sampling interval T_s reduces, the spacing between the sampling frequency ω_s reduces thus bringing the cloned signal spectrums closer. Therefore with the reduced sampling rate there is a distinct possibility that the tail ends of the adjoining signal spectrums may overlap. This overlapping of the spectrums is undesirable because it causes loss of information at the higher frequencies and this phenomenon is called *aliasing.* The problem of aliasing frequently happens in practice.
- The problem of aliasing is removed by sampling the signal at a speed greater than twice the highest frequency in the signal spectrum. This sampling rule is called the *Shanon's sampling theorem.*
- Shanon's sampling theorem also leads to the conclusion that the highest frequency that can be recovered in a digital signal is $\omega_s/2$ which is also called the *Nyquist frequency,* ω_N.

- For an ideal sampling, equation-(2.26) also corresponds to the z-transform and for such a case FT$[x(t)]$ is given by $X(\omega) = \sum_{k=0}^{\infty} x(kT_s)e^{-jk\omega_s T_s}$.

2.5.4 Discrete time periodic signal

This subsection discusses the main objective of translating the discrete signals into discrete frequency points using the DFT. Let the discrete points t in $x(t, N)$ be given by $t = kT_s$, and if this signal is periodic with a period T, then its DFT and its inverse are expressed as [12]:

$$\begin{aligned} X(m\Delta\omega) &= \sum_{k=1}^{N} x(kT_s)e^{-jkm\Delta\omega T_s} \\ x(kT_s) &= \frac{1}{N}\sum_{m=1}^{M} X(m\Delta\omega)e^{jkm\Delta\omega T_s} \end{aligned} \tag{2.27}$$

where $\Delta\omega = 2\pi/T$ is the fundamental frequency and M is the total number of frequency points. The following observations thus hold for periodic discrete signals:

- *FT of a periodic discrete signal results in a discrete frequency function.*
- The discrete frequency function appears with its clones at integral multiples of the sampling frequency ω_s.
- Since the time signal is periodic, the discrete frequency points are separated by $\Delta\omega = 2\pi/T$.

2.5.5 Implementation of DFT (FFT)

Some practical issues that need to be considered for the implementation of DFT are presented in this subsection. The following convention is adopted for the implementation of DFT.

Remark 2.2

- *For a periodic discrete signal $x(t, N)$, each sampling instant is given by $t = kT_s$, however the range of k is set as $0 \leq k \leq N-1$.*
- *Time period of $x(t, N)$ is T where $T = NT_s$.*
- *Frequency points are given by m such that $0 \leq m \leq N-1$ and the frequency increment is $\Delta\omega = 2\pi/T$. Therefore $X(m\Delta\omega)$ represents a discrete frequency function and sometimes for convenience $m\Delta\omega$ is denoted by ω_m, i.e. $X(m\Delta\omega) = X(\omega_m)$.*

- *Let* $W = e^{-j\Delta\omega T_s}$ *and* $W^{-1} = e^{j\Delta\omega T_s}$.

With respect to the above remark-2.2, the DFT pair is given by [12]:

$$\begin{aligned} X(m\Delta\omega) &= \sum_{k=0}^{N-1} x(k)W^{mk} & \Rightarrow \quad & X(m\Delta\omega) = \mathrm{DFT}[x(k)] \\ x(k) &= \frac{1}{N}\sum_{m=0}^{N-1} X(m\Delta\omega)W^{-mk} & \Rightarrow \quad & x(k) = \mathrm{DFT}^{-1}[X(m\Delta\omega)] \end{aligned} \tag{2.28}$$

The frequency points $0 \leq m \leq N-1$ in the above expression span the range $(0, \omega_s)$. Whereas from subsection 2.5.3 it is known that only the frequency range $(0, \omega_N)$ makes sense for discrete systems. Therefore from $X(m\Delta\omega)$, only the range $0 \leq m \leq (N/2) - 1$ that corresponds to $(0, \omega_N)$ represents the true signal spectrum. The remaining half *i.e.* $N/2 \leq m \leq N-1$ corresponds to the adjoining cloned spectrum that can be interpreted to represent the negative time or frequency [12]. The first expression in equation-(2.28) can also be re-written [12] as:

$$X(m\Delta\omega) = \underbrace{\sum_{k=0}^{N-1} x(k)\cos(km\Delta\omega T_s)}_{X_r(m\Delta\omega) \text{ or } X_r(\omega_m)} + j\underbrace{\sum_{k=0}^{N-1} x(kT_s)\sin(km\Delta\omega T_s)}_{X_i(m\Delta\omega) \text{ or } X_i(\omega_m)} \tag{2.29}$$

Assuming $x(t, N)$ to be a sequence of real numbers, the above equation then conveys the following:

- $X_r(\omega_m)$ is even (*i.e.* $X(\omega_s - \omega_m) = X(\omega_m)$) and $X_i(\omega_m)$ is odd (*i.e.* $X(\omega_s - \omega_m) = -X(\omega_m)$).
- If $x(t, N)$ is even (*i.e.* $x(N-k) = x(k)$), then $X(\omega_m) = X_r(\omega_m)$ and it is an even function.
- If $x(t, N)$ is odd (*i.e.* $x(N-k) = -x(k)$), then $X(\omega_m) = X_i(\omega_m)$ and it is an odd function.

Equation-(2.29) shows that N^2 computational steps are required to calculate the DFT. This means that for a large N, which is common in many applications, the DFT computation by using equation-(2.29) would require an enormous amount of computer time, memory and speed. In an effort to reduce the computational burden of DFT, J.W.Cooley and J.W.Tukey in 1965 proposed the Fast Fourier Transform (FFT) algorithm, that reduced the number of computational steps to $N\log_2 N$. The effect of reduction in the computation steps using FFT is felt more when N increases. For example when $N = 64$, the ratio $\frac{N\log_2 N}{N^2}$ is 0.0938, whereas for $N = 2048$ this ratio drops to 0.0054. The FFT algorithm can be applied to any number of data points, but the maximum compuational efficiency is obtained by constraining the data length to 2^N. There are several types of FFT algorithms, but they are not discussed because:

- FFT algorithms are involved and they are beyond the scope of this thesis.
- The objective of this section is only to introduce the basic concepts of FFT.
- Implementation of FFT is quite trivial because the software and hardware is readily available. For example, under the MATLAB© environment[11]; a simple command: *X=fft(x)* will convert a time domain vector x into a frequency domain vector X by using the FFT algorithm.

2.5.6 Data windowing

Windowing of data are used to obtain a smoother and better estimate of the signal spectrum. An important property of DFT mentioned in subsection 2.5.3 that is useful for data windowing is:

- *Multiplication of two time domain signal results in a convolution ($\star$) of their spectrums in the frequency domain and vice-versa.* For example $\text{DFT}[x(t)w(t)] = \frac{1}{N}X(\omega_m) \star W(\omega_m) = \frac{1}{N}\sum_{m=0}^{N-1} X(\omega_m)W(\omega_s - \omega_m)$.

Windows are time domain even functions that are used to trim or shape a large data set with an objective to recover the original signal spectrum from the trimmed or re-shaped data set. An elementary example of a window function is the *rectangular window*. When rectangular window is multiplied with a data whose length is longer than this window, it causes it to abruptly truncate the data. Or, in other words, finite data record can be interpreted as a multiplication of rectangular window of finite length with the original infinitely long data set.

A rectangular window is essentially same as the non-ideal pulse function $\hat{\delta}(t)$, but with a wider base. Therefore a sinc function for the rectangular window drops off more rapidly than for $\hat{\delta}(t)$. The magnitude spectrum[12] of this sinc function shows a main lobe and a series of sidelobes of diminishing magnitude as depicted in Figure 2.6 by the thick chained lines.

It is desired that the main lobe be narrow and the maximum side lobe level be as small as possible for best window performance. But it turns out that both these objectives are contradictory so a compromise is required between reducing the width of main lobe

---

[11] A registered trademark of The Math Works, Inc.

[12] Ideally the magnitude spectrums in Figure 2.6 should be shown in a logarithmic scale, instead they are shown in a linear scale because on a log scale a large number of points are required which exceeds the memory limitation of the graphical software used by the author.

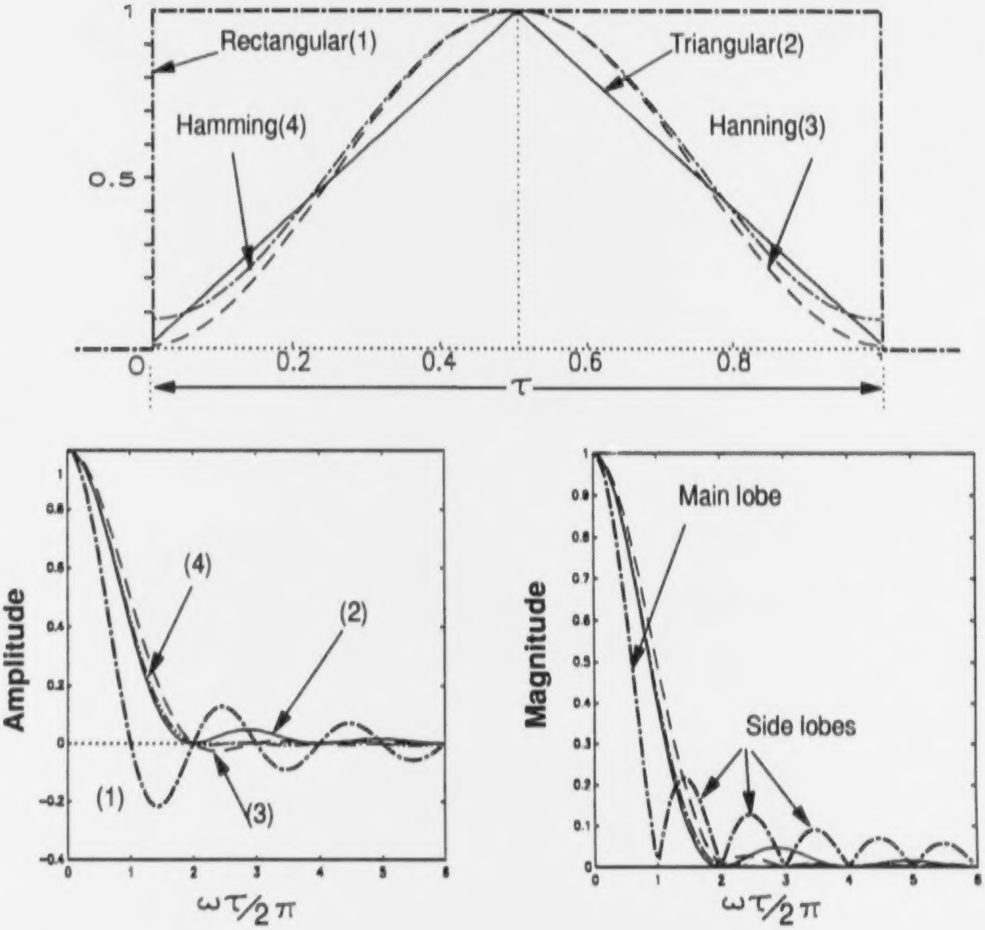


Figure 2.6: Top: Different window functions in time domain. Bottom-left: Amplitude of different windows in frequency domain. Bottom-right: Magnitude of these window functions in frequency domain.

verses having smaller side lobes. Smaller side lobes are desirable because they minimize *leakage* in the spectral estimates and thus help in obtaining better signal spectrum. Similarly narrow main lobe helps to capture sharp features such as resonance peaks in the signal spectrum. The rectangular window gives undesirable side lobe characteristics but its main lobe is narrower. Therefore the use of a rectangular window often gives spurious values at different frequencies but it can also identify sharp features in the signal spectrum better.

Recurence of spurious signals due to rectangular window is a major problem in spectrum estimations. Hence to cope up with this challenge researchers in signal processing have directed their efforts to develop various other type of windows. A partial list of some windows [12] are shown in Table 2.1 and they are graphically illustrated in Figure 2.6 . Figure 2.6 shows that: (a) rectangular window has higher side lobes than any other windows in the figure; (b) the amplitude spectrum of triangular window is positive at all frequencies unlike other windows; and (c) smaller side lobes for Hamming/Hanning windows are realized by compromizing width of the main lobe.

Table 2.1: Time and frequency representation of different windows

Win	Time domain	Frequency domain
1	$\mathrm{w(t)} = 1 \text{ for } \lvert t\rvert < \frac{\tau}{2}$ $= 0 \text{ elsewhere}$	$W(\omega) = \tau \mathrm{sinc}(\frac{\omega\tau}{2})$
2	$\mathrm{w(t)} = 1 - \frac{2\lvert t\rvert}{\tau} \text{ for } \lvert t\rvert \leq \frac{\tau}{2}$ $= 0 \text{ elsewhere}$	$W(\omega) = \frac{\tau}{2}[\mathrm{sinc}(\frac{\omega\tau}{4})]^2$
3	$\mathrm{w(t)} = 0.5 + 0.5\cos(\frac{2\pi t}{\tau}) \text{ for } \lvert t\rvert \leq \frac{\tau}{2}$ $= 0 \text{ elsewhere}$	$W(\omega) = \frac{\tau}{2}[\frac{1}{1-(\frac{\omega\tau}{2\pi})^2}]\mathrm{sinc}(\frac{\omega\tau}{2})$
4	$\mathrm{w(t)} = 0.54 + 0.46\cos(\frac{2\pi t}{\tau}) \text{ for } \lvert t\rvert \leq \frac{\tau}{2}$ $= 0 \text{ elsewhere}$	$W(\omega) = [\frac{0.54-0.08(\frac{\omega\tau}{2\pi})^2}{1-(\frac{\omega\tau}{2\pi})^2}]\tau\mathrm{sinc}(\frac{\omega\tau}{2})$

where: 1=Rectangular, 2=Triangular, 3=Hanning and 4=Hamming windows

Since data windowing is a multiplication of data with window function in the time domain, it translates into a convolution between the window and the signal spectrums in the frequency domain. Therefore depending upon the application and nature of the signal spectrum, it becomes important to select the window functions carefully. For example, chemical processes usually have overdamped responses which are characterized by smooth spectrums. Therefore for such cases Hanning or Hamming windows are adequate for recovering the original signal spectrums. Hanning or Hamming windows are ideally suited for smooth spectrums because these windows have broader main lobes and very small side lobes characteristics.

2.5.7 Power spectrum

A power spectrum is used to characterize a signal or to find the relation between two signals in the frequency domain. An equivalent time domain signal characterization is done by the autocovariance function and similarly a cross covariance function is used to determine relation between the signals as discussed in sections 2.4.1 and 2.4.2.

The power spectrum which is used to characterize a signal is called *auto spectrum* and similarly a *cross spectrum* is used to find the relation between two signals. Auto and cross spectrums are used in system identification to estimate frequency behavior of the plant transfer function. Power spectrums also provide an excellent basis for cross validating mathematical models of dynamic systems.

For a discrete signal $x(t, N \rightarrow \infty)$ let the true auto spectrum be denoted by $\Phi_{xx}(\omega)$ and similaly let $\Phi_{yu}(\omega)$ denote the true cross spectrum between $y(t, N \rightarrow \infty)$ and $u(t, N \rightarrow \infty)$. Discrete auto and cross power spectrums are then defined as [10, 13]:

$$\begin{aligned}\Phi_{xx}(m\Delta\omega) &= \sum_{k=-\infty}^{\infty} \gamma_{xx}(k) W^{mk} \\ \Phi_{yu}(m\Delta\omega) &= \sum_{k=-\infty}^{\infty} \gamma_{yu}(k) W^{mk}\end{aligned} \qquad (2.30)$$

The concept of normalization as in case of acf or ccf also applies to the power spectrums to facilitate comparison between different signals. Normalized power spectrums are called ***spectral densities***. Thus the auto and cross spectral densities can be defined as [10]:

$$\begin{aligned}\underline{\Phi}_{xx}(m\Delta\omega) &= \frac{1}{\gamma_{xx}(0)} \sum_{k=-\infty}^{\infty} \gamma_{xx}(k) W^{mk} = \sum_{k=-\infty}^{\infty} \rho_{xx}(k) W^{mk} \\ \underline{\Phi}_{yu}(m\Delta\omega) &= \frac{1}{\gamma_{yu}(0)} \sum_{k=-\infty}^{\infty} \gamma_{yu}(k) W^{mk} = \sum_{k=-\infty}^{\infty} \rho_{yu}(k) W^{mk}\end{aligned} \qquad (2.31)$$

The following remarks apply to power spectrums and also spectral densities:

Remark 2.3

- $\Phi_{xx}(\omega)$ *or* $\Phi_{yu}(\omega)$ *generically denotes spectral values for all frequencies i.e.* $\omega \in [0, \omega_N]$.
- $\Phi_{xx}(\omega)$ *is essentially a magnitude spectrum and therefore it is an even function.*
- $\Phi_{yu}(\omega)$ *is a spectrum of complex numbers and it is an odd function. The cross spectrum therefore contains information about the phase lag or lead between two sets of data and it can also be expressed [10] as:*

$$
\begin{aligned}
\Phi_{yu}(\omega) &= \underbrace{\Phi_{yu_r}(\omega)}_{\text{co-spectra}} + j \overbrace{\Phi_{yu_i}(\omega)}^{\text{quadrature-spectra}} \\
\text{or} & \\
\Phi_{yu}(\omega) &= \underbrace{|\Phi_{yu}(\omega)|}_{\text{magnitude}} \overbrace{\angle\Phi_{yu}(\omega)}^{\text{phase}} \qquad (2.32)
\end{aligned}
$$

In most applications the data length N is limited, therefore the concepts of sample acf and ccf also apply correspondingly to the spectral estimates. Sample auto and cross spectrums are denoted by $\hat{\Phi}_{xx}(\omega)$ and $\hat{\Phi}_{yu}(\omega)$ respectively and they can be similarly defined as in equation-(2.31) by setting finite limits for the summation sign.

Sample spectrums are also called ***periodograms*** and naturally these periodograms are estimates of the true power spectrums. In time domain, sample auto/cross covariance functions provide consistent asymptotically unbiased estimate of the true auto/cross covariance functions. However their FTs *i.e.* sample auto/cross spectrums are not asymptotically consistent estimate of their true power spectrums because their variances never approach zero with the increase in N. The variances in periodograms can be reduced by using window functions and other spectral smoothing techniques which is discussed in subsection 2.5.8.

With the advent of FFT algorithm, it is no longer required to compute the DFT of the auto/cross covariance functions to obtain periodograms. The FFT can directly be applied to the data to compute these periodograms [1] as shown:

$$U(\omega) = FFT[U(t,N)] \quad \text{and} \quad Y(\omega) = FFT[y(t,N)] \qquad (2.33)$$

then

$$
\begin{aligned}
\hat{\Phi}_{uu}(\omega) &= \frac{1}{N} U(\omega) U^H(\omega) = \frac{1}{N} |U(\omega)|^2 \\
\hat{\Phi}_{yu}(\omega) &= \frac{1}{N} Y(\omega) U^H(\omega) \qquad (2.34)
\end{aligned}
$$

where $U^H(\omega)$ is the complex conjugate transpose of $U(\omega)$.

For causally related data sets (*e.g.* in Figure 2.1 $y(t,N)$ is causally related to $u(t,N)$ via $G(z)$), the correlation between the input and the output at a frequency (ω) is described by the ***squared coherency spectrum*** $\kappa^2_{uy}(\omega)$ as:

$$\kappa^2_{uy}(\omega) = \frac{|\Phi_{yu}(\omega)|^2}{\Phi_{uu}(\omega)\Phi_{yy}(\omega)} \qquad (2.35)$$

Concept of sample spectrum or periodograms also applies to squared coherency and it is called ***sample squared coherency spectrum***: $\hat{\kappa}^2_{yu}(\omega)$. The value of squared coherency varies between 1 and 0, signifying complete and no coherency respectively, between input and output. Often the process output is corrupted by noise as shown in Figure 2.1, in such cases squared coherency can be used to estimate the degree to which the output is effected by noise. For example if the noise level is high, a squared coherency will have a lower value and for lower noise level it will have values close to 1. More recently attempts are also being made to use the squared coherency function to acertain the level of non-linearity in the systems [14].

It should be noted that direct substitution of equation-(2.34) in equation-(2.35) will always lead to $\kappa_{uy}^2(\omega) = 1$ even when $y(t)$ is corrupted by noise; which of course will lead to a misleading result. A true picture emerges only when equation-(2.35) is used with spectral smoothing techniques described in the following subsection.

2.5.8 Spectral smoothing

Since the periodograms are not consistent estimate of the spectrum and also increasing N provides no relief, it becomes important to consider modifications in spectral estimation procedures that will yield better results. The procedures by which the signal spectrums are modified to yield results are called *spectral smoothing techniques*. Popular spectral smoothing techniques include periodogram averaging as proposed by Bartlett and data windowing and their combinations as suggested by Welch.

Bartlett's smoothing

Bartlett suggested dividing N point data into K units of M samples each to compute K periodograms and then take their average. Using statistical techniques it is shown that such an averaging naturally leads to convolution between a triangular window and the original spectrum [10]. Furthermore, with increasing K, the variance of the resulting averaged periodogram decreases and thus it becomes a consistent estimate of the true spectrum [1]. However the disadvantage of increasing K is that it reduces the spectral resolution *e.g.* it becomes difficult to identify if the spectrum has a narrow or a sharp peak. Therefore depending on the application a compromise is required between spectral resolution and reduction in the spectral variance.

Window smoothing

Another way to smooth a signal spectrum is to convolve it with an appropriate window function. In case of Bartlett's method, the averaging is equivalent equal to the triangular window. But if a window of desirable characteristics *i.e.* with lower side lobes (*e.g.* Hamming/Hanning windows) is used instead of triangular window; a better compromise between resolution and variance in the periodogram can be achieved. It should be noted that periodograms are non-negative functions and Bartlett's method (*i.e.* triangular window) always result in periodograms with positive values. But the same cannot be said about Hamming or Hanning windows as sometimes these windows can give negative values for the periodograms as at certain frequencies their amplitude shows negative values [1] as can be seen in Figure 2.6.

Welch's smoothing

Welch proposed a combination of spectral averaging and data windowing to achieve desirable results. Welch's method essentially brings together the desirable features of both data windowing and Bartlett's method. Welch's method can be summarized by the following steps [1]:

1. Split the data $x(t,N)$ into K sections of M points in each section. The K data segments can also be overlapped if required.
2. Each data segment is separately multiplied by window function $w(t)$.
3. The FFT of each windowed data segment are computed to obtain K modified periodograms.
4. The resulting periodogram is the average of K periodograms as shown below:

$$\bar{\Phi}_{xx}(\omega_m) = \frac{1}{K}\frac{1}{M\|w\|^2}\sum_{k=1}^{K}\{|X_k^w(\omega_m)|\} \qquad \text{for } 0 \le m \le M-1 \tag{2.36}$$

where

$$\|w\|^2 = \frac{1}{M}\sum_{l=0}^{M-1} w^2(l) \tag{2.37}$$

and $X_k^w(\omega) = \text{FFT}[x(t,M)_k w(t)]$ and where $x(t,M)_k$ is the k^{th} data segment in $x(t,N)$ for $k = 1,2,\ldots,K$. In equation-(2.36) $\bar{\Phi}_{xx}(\omega)$ denotes the average periodogram and the frequency points m span the range $(-\omega_N, \omega_N)$. The average cross periodogram can be determined similarly.

2.5.9 Estimation of frequency response of a transfer function

A major application of signal processing methods in process control is to estimate the frequency behavior of a transfer function for either model validation or controller design. If $y(t,N)$ be causally related to $u(t,N)$ *via* $G(z)$, then in frequency domain this translates into [15]:

$$Y(\omega) = G(\omega)U(\omega) \tag{2.38}$$

and similarly

$$Y^H(\omega) = G^H(\omega)U^H(\omega) \tag{2.39}$$

post-multiplying both side of equation-(2.38) by $U^H(\omega)$ and subsequently using equation-(2.39) results in:

$$Y(\omega)U^H(\omega) = G^H(\omega)U(\omega)U^H(\omega) \tag{2.40}$$

combining equation-(2.34) with the above equation gives:

$$G(\omega) = \frac{\Phi_{yu}(\omega)}{\Phi_{uu}(\omega)} \tag{2.41}$$

similarly multiplying both sides of equation-(2.38) with $G^H(\omega)U^H(\omega)$ and substituting equations-(2.39) and (2.34) yields [15]:

$$|G(\omega)| = \sqrt{\frac{\Phi_{yy}(\omega)}{\Phi_{uu}(\omega)}} \tag{2.42}$$

Equation-(2.41) is very useful because it gives spectrums of both magnitude and phase of the system from time domain process data. Equation-(2.42) can be used when information on the magnitude spectrum alone is required.

2.6 Statistical analysis in signal processing

One of the main objectives of statistical analysis in signal processing is to determine the confidence bounds on the estimated spectrum. Confidence bounds are derived on the assumption that the error or noise $x(t, N)$ have a Gaussian distribution *i.e.* $f_G(x) = \frac{1}{\sqrt{2\pi}} \exp \frac{((x-\mu)/\sigma)^2}{2}$, to which the statistical methods are applied. A portion of this section is devoted to some statistical techniques that are required to estimate the confidence bounds for the signal spectrum.

A measure of spread of the normally distributed population (*i.e.* $x(t, N \to \infty)$) is variance $\sigma^2 = \mathcal{E}[(x-\mu)^2]$, similarly for a finite sample size N, the sample variance is given by:

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{k=1}^{N}(x(k) - \bar{x})^2 \tag{2.43}$$

Normally distributed data for a population (*i.e.* $N \to \infty$) is denoted by $\mathcal{N}(\mu, \sigma^2)$ and for finite N it is similarly given by $\mathcal{N}(\bar{x}, \hat{\sigma}^2)$ (In the last Sections 2.4 and 2.5 this distinction was not made for reasons of simplicity).

χ^2 distribution

The key distribution that is used to determine the spectral bounds is χ^2. For $x(t,N) \in \mathcal{N}(0,1)$, the sum of its squared elements is said to have a χ^2_N distribution with a degree of freedom (*dof*) N *i.e.* [10]:

$$\chi^2_N = x_1^2 + x_1^2 + \cdots + x_N^2 \tag{2.44}$$

For the general case if ν denotes the dof, then χ^2_ν is given by [10]

$$f_{\chi^2_\nu}(x) = \frac{1}{2^{\frac{\nu}{2}}\Gamma(\frac{\nu}{2})} x^{(\frac{\nu}{2})-1} e^{-\frac{x}{2}} \tag{2.45}$$

for $0 \leq x \leq \infty$ and where $\Gamma(\nu/2) = \int_0^\infty e^{-t} t^{(\nu/2)-1} dt$. If $x(t,N) \in \mathcal{N}(\mu, \sigma^2)$, then it is shown [10] that:

$$(N-1)\frac{\hat{\sigma}^2}{\sigma^2} \in \chi^2_{(N-1)} \tag{2.46}$$

i.e. the dof for the ratio between the sample and true variance is $\nu = N-1$. As $\nu \to \infty$, $\chi^2_\nu \to \mathcal{N}(\mu, \sigma^2)$ (Central limit theorem), however for smaller dof's, χ^2_ν is vastly different from $\mathcal{N}(\mu, \sigma^2)$. Probability limits of $\nu\hat{\sigma}^2/\sigma^2$ are [10]:

$$\mathcal{P}\left[\chi^2_\nu\left(\frac{\alpha}{2}\right) < \frac{\nu\hat{\sigma}^2}{\sigma^2} \leq \chi^2_\nu\left(1-\frac{\alpha}{2}\right)\right] = 1-\alpha \tag{2.47}$$

where α is the confidence limit. Equation-(2.47) can also be rearranged to:

$$\mathcal{P}\left[\frac{\nu}{\chi^2_\nu(1-\alpha/2)} < \frac{\sigma^2}{\hat{\sigma}^2} \leq \frac{\nu}{\chi^2_\nu(\alpha/2)}\right] = 1-\alpha \tag{2.48}$$

Some properties of χ^2_ν are [10]:

$$\begin{aligned} \mathcal{E}[\chi^2_\nu] &= \nu \\ \mathrm{var}(\chi^2_\nu) &= 2\nu \\ \chi^2_\nu &= \chi^2_{\nu_1} + \chi^2_{\nu_2} + \cdots + \chi^2_{\nu_n} \end{aligned} \tag{2.49}$$

where $\nu = \nu_1 + \nu_2 + \cdots + \nu_n$. If $x \in a{\chi_\nu}^2$, where a and ν are unknown, then using equation-(2.49) these two unknown parameters can be shown to be:

$$\begin{aligned} \nu &= \frac{2(\mathcal{E}[x])^2}{\mathrm{var}(x)} \\ a &= \frac{\mathcal{E}[x]}{\nu} \end{aligned} \tag{2.50}$$

because the first two moments of x are $\mathcal{E}[x] = a\nu$ and $\mathrm{var}[x] = 2a^2\nu$. The above equation-(2.50) will be used later to describe the statistical distribution of periodograms.

Fisher's F distribution

Related to χ^2_ν there is yet another distribution which is called the *Fisher's F distribution*. The F distribution is applied to estimate the bounds on magnitude and phase spectrums. It compares the variances of two unrelated normal data sets of different lengths (N_1 and N_2) having distributions $\mathcal{N}(\mu_1, {\sigma_1}^2)$ and $\mathcal{N}(\mu_2, {\sigma_2}^2)$. Therefore by using equation-(2.46), with the F distribution as defined as [10]:

$$F_{\nu_1\nu_2} = \frac{\hat{\sigma_1}^2/{\sigma_1}^2}{\hat{\sigma_2}^2/{\sigma_2}^2} = \frac{\chi^2_{\nu_1}/\nu_1}{\chi^2_{\nu_2}/\nu_2} \tag{2.51}$$

If the values of $F_{\nu_1\nu_2}$ distribution are given by $f_{\nu_1\nu_2}(1-\alpha)$, then for a fixed confidence limit of $(1-\alpha)$, plots of $f_{\nu_1\nu_2}(1-\alpha)$ are available for different ν_1 and ν_2. The corresponding confidence level on $F_{\nu_1\nu_2}$ is given in [10] as:

$$\mathcal{P}\left[\frac{{\sigma_2}^2\hat{\sigma_1}^2}{{\sigma_1}^2\hat{\sigma_2}^2} \leq f_{\nu_1\nu_2}(1-\alpha)\right] = 1-\alpha \tag{2.52}$$

It may also be noted that $F_{\nu_1\nu_2} = 1/F_{\nu_2\nu_1}$, therefore $f_{\nu_1\nu_2}(1-\alpha)$ and $f_{\nu_2\nu_1}(1-\alpha)$ can be used to construct confidence intervals as shown in subsection 2.6.3.

2.6.1 Confidence bounds on smoothened auto spectrums

In practice N is finite and the data set is often corrupted by noise which tends to adversely affect the spectral estimates discussed in Sections 2.5.6 to 2.5.8. Therefore uncertainties in spectral estimates are always accounted for by the confidence bounds. Often these spectral uncertainties are assumed to be stochastic, and therefore statistical techniques outlined in Section 2.6 are used to compute the bounds. A number of assumptions and approximations are made for deriving these spectral bounds. An important relation used in this context is the *Parseval's theorem*[13] [1]:

$$\sum_{m=0}^{M-1} |X(m\Delta\omega)|^2 = \sum_{k=0}^{N-1} x^2(k) \tag{2.53}$$

If $x(t, N \to \infty) \in \mathcal{N}(0, \sigma^2)$, then from Parseval's relation (2.53) and equation-(2.43) it is clear that:

$$\sigma^2 = \frac{1}{N}\sum_{k=0}^{N-1} x(k)^2 = \frac{1}{N}\sum_{m=0}^{M-1} |X(m\Delta\omega)|^2 = \sum_{m=0}^{M-1} \Phi_{xx}(m\Delta\omega) \tag{2.54}$$

Equation-(2.54) is validated by illustrations in Figure 2.7 for the example $x(t, N) \in \mathcal{N}(0, 1.0354)$, where it is shown that good estimates of $\hat{\sigma}^2$ are obtained from the signal spectrum. Subsets of Figure 2.7 further show that: (a) an unsmoothened spectrum exhibits

---

[13] The Parsevals theorem can be expressed in several forms. Some other forms of Parseval's theorem are discussed in Chapters 4 and 7.

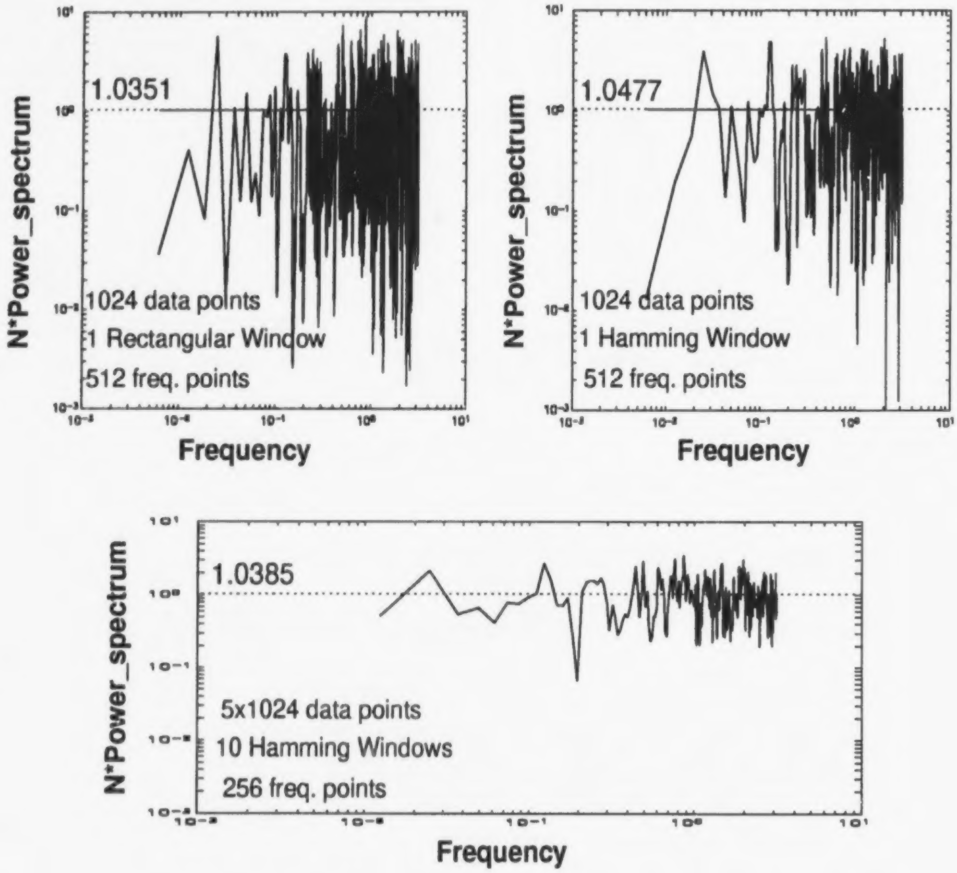


Figure 2.7: Use of periodogram to estimate the variance of a white noise signal.

more variations but it yields a good mean value, (b) improper smoothing (*i.e.* use of just one window) leads to biased estimate and (c) proper smoothing reduces variance in the estimated spectrum without compromising the mean value. The statistical distribution of sample spectrum of white noise is shown to be:

$$\frac{2\hat{\Phi}_{xx}(\omega)}{\sigma^2} = \frac{2}{N\sigma^2}({X_r}^2(\omega) + {X_i}^2(\omega)) \in \chi_2^2 \tag{2.55}$$

Similarly the above spectral distribution ***approximately*** holds for the filtered white noise *i.e.* $x(t) = G(z)\xi(t)$ also.

It is proved in the literature that $\text{var}(\hat{\Phi}_{xx}(\omega))$ increases with the increase in N contrary to the expectation that it should decrease. Therefore periodograms are not a consistent estimator of the true spectra but this situation is rectified by resorting to spectral smoothing techniques described in subsection 2.5.8. Smoothened periodograms have distribution similar to equation-(2.55), but they have more dof ν *i.e.* $\nu\bar{\Phi}_{xx}(\omega)/\Phi_{xx}(\omega) \in a{\chi_\nu}^2$ and it is shown that [10]:

$$\begin{aligned} \mathcal{E}[\bar{\Phi}_{xx}(\omega)] &\approx \Phi_{xx}(\omega) \\ \text{var}(\bar{\Phi}_{xx}(\omega)) &\approx \frac{1}{N}\Phi_{xx}^2(\omega)\sum_{k=-\omega_N}^{\omega_N} W^2(\omega) = \frac{\Phi_{xx}^2(\omega)}{N}\sum_{k=0}^{M-1} w^2(k) \end{aligned} \tag{2.56}$$

The above expressions-(2.56) show: (a) with increasing N, spectral variance of smoothened periodogram decreases because N is in the denominator; and (b) the spectral variance is weighted by the dof of the window function. Parsevals relation is used to calculate the spectral variance in the above equation. From equations-(2.50) and (2.56) it is shown that:

$$\nu = \frac{2N}{\sum_{k=0}^{M-1} w^2(k)} \tag{2.57}$$

Therefore using equations-(2.47), the confidence bound for $\nu\bar{\Phi}_{xx}(\omega)/\Phi_{xx}(\omega)$ follows:

$$\mathcal{P}\left[\chi_\nu^2\left(\frac{\alpha}{2}\right) < \frac{\nu\bar{\Phi}_{xx}(\omega)}{\Phi_{xx}(\omega)} \le \chi_\nu^2\left(1 - \frac{\alpha}{2}\right)\right] = 1 - \alpha \tag{2.58}$$

2.6.2 Confidence bounds on the smoothened cross spectrum

Derivation of confidence bounds for the cross spectrum is more involved and it is beyond the scope of this thesis hence only some final results are presented. The bounds for cross spectrum are provided separately for $|\bar{\Phi}_{yu}(\omega)|$, $\bar{\kappa}_{yu}^2$ and $\bar{\varphi}_{yu}(\omega)$ and their variances are given by [10]:

$$
\begin{aligned}
\mathrm{var}(|\bar{\Phi}_{yu}(\omega)|) &\approx \frac{(\mathcal{E}[|\bar{\Phi}_{yu}(\omega)|])^2}{\nu}\left(1+\frac{1}{\kappa_{yu}^2(\omega)}\right) \\
\mathrm{var}(\bar{\kappa}_{yu}^2(\omega)) &\approx \frac{4\kappa_{yu}^2(\omega)}{\nu}(1-\kappa_{yu}^2(\omega))^2 \\
\mathrm{var}(\bar{\varphi}_{yu}(\omega)) &\approx \frac{1}{\nu}\left(\frac{1}{\kappa_{yu}^2(\omega)}-1\right)
\end{aligned}
\tag{2.59}
$$

The above set of expressions (equation-(2.59)) convey that:

- Variances of these estimates are dependent on spectral smoothing *i.e.* dof ν and $\kappa_{yu}^2(\omega)$.
- For $\kappa_{yu}^2(\omega) = 1$ (*i.e.* no noise) there are no variances in $\bar{\kappa}_{yu}^2(\omega)$ and $\bar{\varphi}_{yu}(\omega)$.
- For $\kappa_{yu}^2(\omega) \to 0$ (*i.e.* no input-output coherency), $\mathrm{var}(|\bar{\Phi}_{yu}(\omega)|) \to \infty$ and $\mathrm{var}(\bar{\varphi}_{yu}(\omega)) \to \infty$.
- Effect of $\kappa_{yu}^2(\omega)$ on these variances is more profound than the smoothing factor ν.

$|\hat{\Phi}_{yu}(\omega)|$ has larger variance than the component signals y and u because the distribution of $|\hat{\Phi}_{yu}(\omega)|$ is approximately the product of χ_2^2 distributions of y and u. The confidence bounds on only the squared coherency spectrum is discussed because they provide more useful information than the amplitude or phase of the cross spectrum. The 95% confidence bound on $\bar{\kappa}_{yu}^2(\omega)$ is given by:

$$
\underline{\bar{\kappa}'}(\omega) = (\kappa^U(\omega), \kappa^L(\omega)) = \underbrace{\frac{1}{2}\ln\frac{1+\bar{\kappa}_{yu}(\omega)}{1-\bar{\kappa}_{yu}(\omega)}}_{\mathrm{arctanh}(\bar{\kappa}_{yu}(\omega))} \pm 1.96\sqrt{\frac{1}{\nu}}
\tag{2.60}
$$

where $\underline{\bar{\kappa}'}$ or ($\kappa^U(\omega)$ and $\kappa^L(\omega)$) are the upper and lower bounds in terms of transformed coordinates *i.e.* $\mathrm{arctanh}(\bar{\kappa}_{yu}(\omega))$. The bounds on the original coordinate system are obtained by the following inverse transform (*i.e.* $\mathrm{arctanh}^{-1}$):

$$
\underline{\bar{\kappa}}(\omega) = \underbrace{\frac{e^{2\underline{\bar{\kappa}'}(\omega)}-1}{e^{2\underline{\bar{\kappa}'}(\omega)}+1}}_{\mathrm{arctanh}^{-1}(\underline{\bar{\kappa}'}(\omega))}
\tag{2.61}
$$

where $\underline{\bar{\kappa}}(\omega) = (\kappa_{yu}^U(\omega), \kappa_{yu}^L(\omega))$ *i.e.* $\kappa_{yu}^L(\omega) \leq \bar{\kappa}_{yu}(\omega) \leq \kappa_{yu}^U(\omega)$. The confidence bounds on the magnitude and phase of $\hat{\Phi}_{yu}(\omega)$ are not mentioned here because it makes more sense to discuss them in the context of estimating the transfer functions as discussed in the following subsection 2.6.3.

2.6.3 Confidence bound on smoothened gain and phase spectrums

In order to calculate the confidence bounds on the gain and phase spectrums, it is assumed that the output $y(t)$ is corrupted by white noise $\xi(t)$. Therefore letting $v(t) = \xi(t)$ in equation-(2.1), the frequency response of the plant becomes

$$Y(\omega) = G(\omega)U(\omega) + Z(\omega) \tag{2.62}$$

where $Z(\omega) = DFT[\xi(t, N)]$. If $G_M(\omega)$ represents the spectral estimate of $G(\omega)$, then equation-(2.62) can also be written as:

$$Y(\omega) = G_M(\omega)U(\omega) + \hat{Z}(\omega) \tag{2.63}$$

where $\hat{Z}(\omega)$ is an estimate of $Z(\omega)$. The following assumptions hold for equation-(2.63):

Assumption 2.1

- *$u(t, N) \in \mathcal{N}(0, \sigma_u^2)$, $\xi(t, N) \in \mathcal{N}(0, \sigma_\xi^2)$ and $\hat{\xi}(t, N) \in \mathcal{N}(0, \sigma_{\hat{\xi}}^2)$.*
- *$\rho_{u\xi}(k) = 0$ and $\rho_{u\hat{\xi}}(k) = 0$ for all k.*

Equations-(2.62) and (2.63) are equated to replace $Y(\omega)$ to obtain

$$Z(\omega) = \hat{Z}(\omega) + U(\omega)(G_M(\omega) - G(\omega)) \tag{2.64}$$

Multiplying equation-(2.64) by $Z^H(\omega)$ and use of assumption 2.1 leads to:

$$\Phi_{\xi\xi}(\omega) = \Phi_{\hat{\xi}\hat{\xi}}(\omega) + \Phi_{uu}(\omega)|G_M(\omega) - G(\omega)|^2 \tag{2.65}$$

The following expression results after: (a) substituting smoothened periodograms in the preceding equation-(2.65); and (b) multiplying the resultant equation by $\nu/\Phi_{ZZ}(\omega)$:

$$\frac{\nu\bar{\Phi}_{\xi\xi}(\omega)}{\Phi_{\xi\xi}(\omega)} = \frac{\nu\bar{\Phi}_{\hat{\xi}\hat{\xi}}(\omega)}{\Phi_{\xi\xi}(\omega)} + \frac{\nu\bar{\Phi}_{uu}(\omega)}{\Phi_{\xi\xi}(\omega)}|G_M(\omega) - G(\omega)|^2 \tag{2.66}$$

Equation-(2.66) conveys the following:

- LHS term has a χ^2_ν distribution.
- LHS is a result of two statistically independent terms shown in the RHS.
- It is shown in literature [10, 5] that the first and second RHS terms have $\chi^2_{\nu-2}$ and χ^2_2 distributions respectively.

- In view of the above observations, ratio of the two RHS terms can be taken to obtain the F distribution.

For the sake of derivation of confidence bounds, it is assumed that y and u are uncorrelated *i.e.* $G(\omega) = 0$ [10], then equations-(2.35), (2.41) and (2.42) can be applied to the terms on the RHS to obtain: $\nu\bar{\Phi}_{uu}(\omega)\bar{G}_M^2(\omega)/\Phi_{\xi\xi}(\omega) = \nu\bar{\Phi}_{yy}^2(\omega)\bar{\kappa}_{yu}^2(\omega)/\Phi_{\xi\xi}(\omega)$ and $\nu\bar{\Phi}_{\hat{\xi}\hat{\xi}}(\omega)/\Phi_{\xi\xi}(\omega) = \nu\bar{\Phi}_{yy}(\omega)(1-\bar{\kappa}_{yu}^2(\omega))/\Phi_{\xi\xi}(\omega)$ and subsequently taking their ratio gives [10]:

$$\frac{(\nu-2)\bar{\kappa}_{yu}^2(\omega)}{2(1-\bar{\kappa}_{yu}^2(\omega))} \in F_{2,\nu-2} \tag{2.67}$$

therefore from equations-(2.52), (2.66) and (2.67) it is clear that ratio of RHS terms in equation-(2.66) will have the following confidence interval [10]:

$$\mathcal{P}\left[\frac{\nu-2}{2}\frac{\bar{\Phi}_{uu}(\omega)|\bar{G}_M(\omega)-G(\omega)|^2}{\bar{\Phi}_{\hat{\xi}\hat{\xi}}(\omega)} \leq f_{2,(\nu-2)}(1-\alpha)\right] = 1-\alpha \tag{2.68}$$

By rearranging equation-(2.68) and using equations-(2.66) and (2.67) gives the following confidence intervals for gain and phase spectrums [10]:

$$\begin{aligned}(G_M^U(\omega), G_M^L(\omega)) &= \bar{G}_M(\omega)\left\{1 \pm \sqrt{\frac{2}{\nu-2}f_{2,(\nu-2)}(1-\alpha)\left(\frac{1-\bar{\kappa}_{yu}^2(\omega)}{\bar{\kappa}_{yu}^2(\omega)}\right)}\right\} \\ (\varphi_M^U(\omega), \varphi_M^L(\omega)) &= \bar{\varphi}_M(\omega) \pm \sin^{-1}\sqrt{\frac{2}{\nu-2}f_{2,(\nu-2)}(1-\alpha)\left(\frac{1-\bar{\kappa}_{yu}^2(\omega)}{\bar{\kappa}_{yu}^2(\omega)}\right)}\end{aligned} \tag{2.69}$$

The confidence intervals presented in this section has a link with the least squares estimates which is a subject of discussion in Chapter 3. Therefore further discussions on confidence intervals are deferred to Chapter 3.

2.7 Case studies

The purpose of this section is to highlight the application of signal processing methods described in Sections 2.5.5 to 2.5.9 and 2.6.3 via illustrative examples. These examples show that the outcome of signal processing methods depend on spectral smoothing techniques, experiment design and the level of noise in the data. With a careful choice of design parameters such as input design and spectral smoothing, reliable estimates of the spectrum can be obtained.

The use of signal processing methods is illustrated by applying them to estimate the frequency response (Bode plots) of the following plant from its input-output data record (equation-(2.17)):

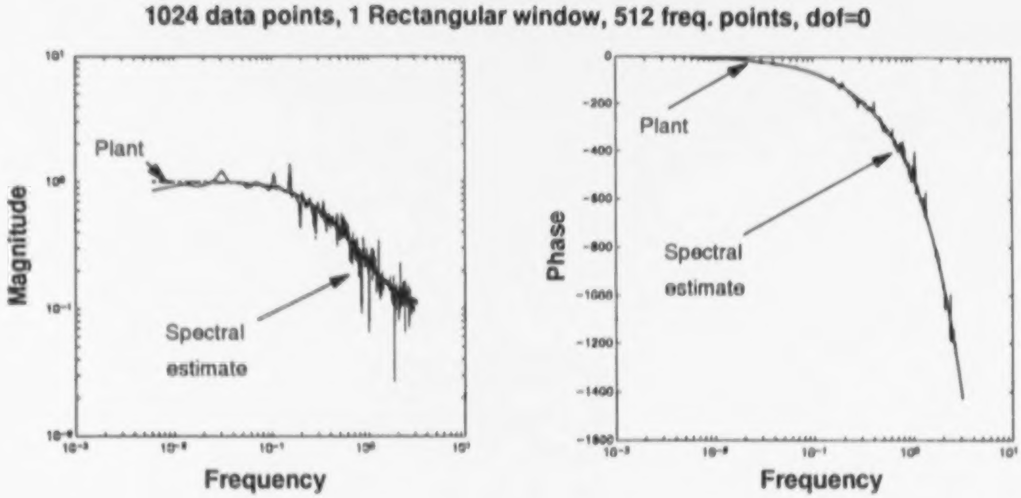


Figure 2.8: Estimation of Bode plots for a noise free plant using (i) white noise excitation and (ii) no spectral smoothing. Left: Magnitude spectrum. Right: Phase spectrum.

$$G_1(q^{-1}) = \frac{0.2q^{-8}}{1 - 0.8q^{-1}} \tag{2.70}$$

The effectiveness of spectral estimation techniques are shown by considering three cases; namely: (a) noise-free data; (b) output corrupted by white measurement noise (SNR=3.03); and (c) output corrupted by ARMA noise (SNR=3.0). The input-output data for case-a has already been shown in Figure 2.5. For other cases the output record will be shown subsequently. The case of noise-free data are of course the simplest one, but it is considered to show the effect of spectral smoothing methods on the spectral estimations. The level of noise considered (*i.e.* $SNR \approx 3.0$) in cases (a) and (b) is quite significant and it is shown how signal processing methods yield reliable results for such a level of noise.

2.7.1 Case: Noise free plant

Case-1: Figure 2.8 shows estimated magnitude and phase spectrums match well with the true Bode plots of the plant. Since no spectral smoothing is employed: (a) estimated spectrums show significant variance even though the plant is noise free; and (b) no spectral bounds are calculated because dof=0.

Case-2: The case presented in Figure 2.9 is similar to the previous one, except that only 1 Hanning window is used to shape the data. This results in: (a) reduced spectral variance without compromising the accuracy; and (b) shows confidence bounds and they are depicted as UB and LB signifying upper and lower bounds respectively. The LB is so

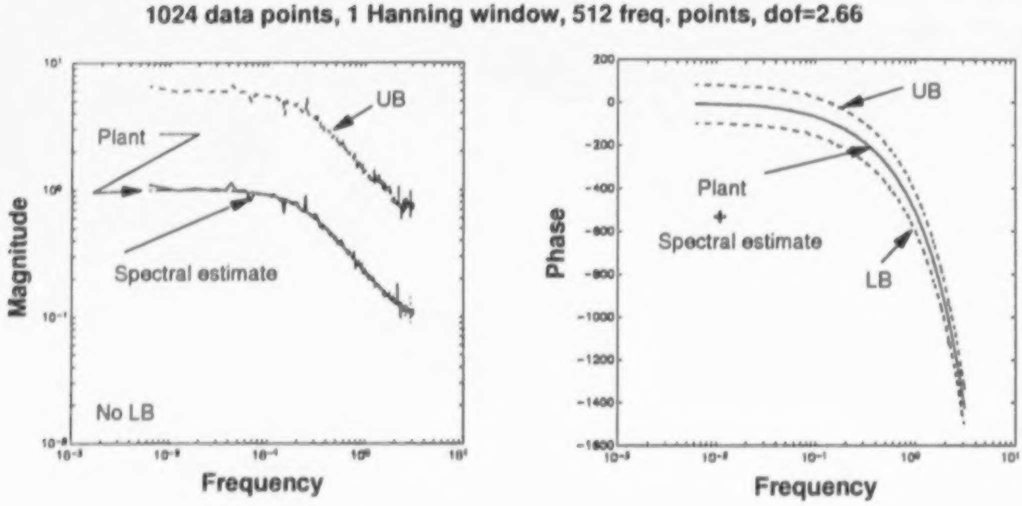


Figure 2.9: Estimation of Bode plots for a noise free plant with bounds using (i) white noise excitation and (ii) 1 Hanning window of length $M = 1024$. Left: Magnitude spectrum. Right: Phase spectrum.

small for the magnitude spectrum because of small dof, that it is beyond the scale of the graph.

Case-3: The effect of Welch's spectral smoothing method in Figure 2.10 shows: (a) further reduction in the variance of the estimated spectrums without compromising the accuracy; (b) tighter confidence bounds owing to higher dof; but (c) with a reduced frequency span (shown by the vertical dotted lines) because of data segmentation.

Case-4: The reduced frequency span is shown to increase suitably by increasing the data length as depicted in Figure 2.11. The tightness of confidence bounds is same as in the previous case because there is no apparent change in the value of dof. However for this case the variance in the estimated spectrum seems to increase due to increased data length.

Case-5: Previous cases show that white noise excitation always result in some variance in the estimated spectrums, irrespective of the smoothing effort. Figure 2.12 on the other hand shows that a low frequency square type excitation yield very smooth spectrums without compromising the quality of estimation. It may however be noted that: (a) spectral smoothing methods are still required to ensure the reliability of spectral estimates; and (b) it is shown later in Section 2.8 that square type excitation does not work well when the data are corrupted by noise.

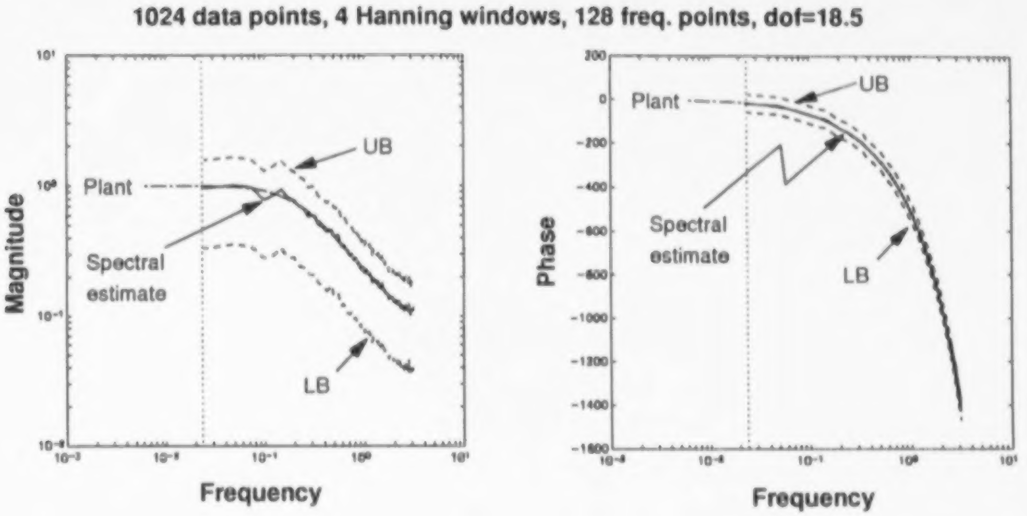


Figure 2.10: Estimation of Bode plots for a noise free plant with upper and lower bounds using (i) white noise excitation and (ii) 4 Hanning windows of length $M = 256$ each. Left: Magnitude spectrum. Right: Phase spectrum.

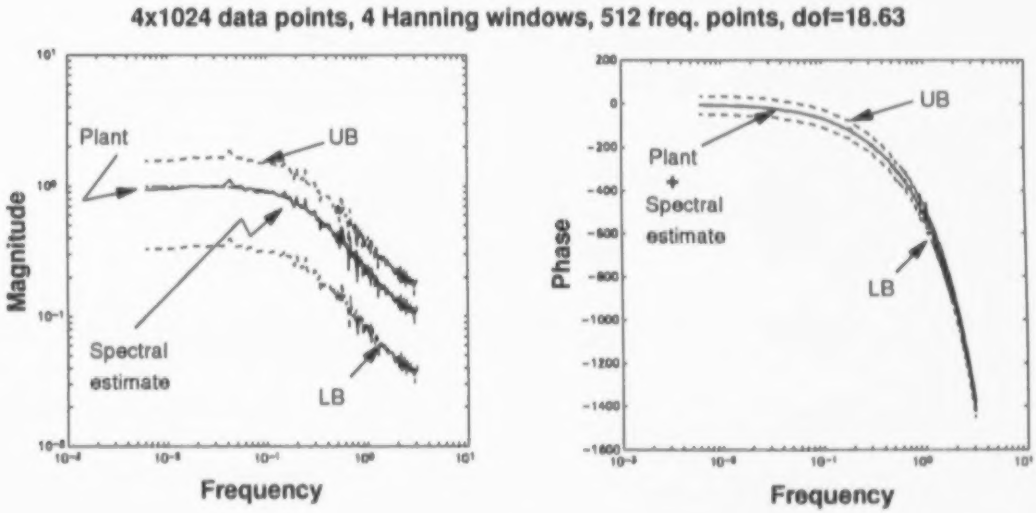


Figure 2.11: Estimation of Bode plots for a noise free plant with upper and lower bounds using (i) white noise excitation and (ii) 4 Hanning windows of length $M = 1024$ each. Left: Magnitude spectrum. Right: Phase spectrum.

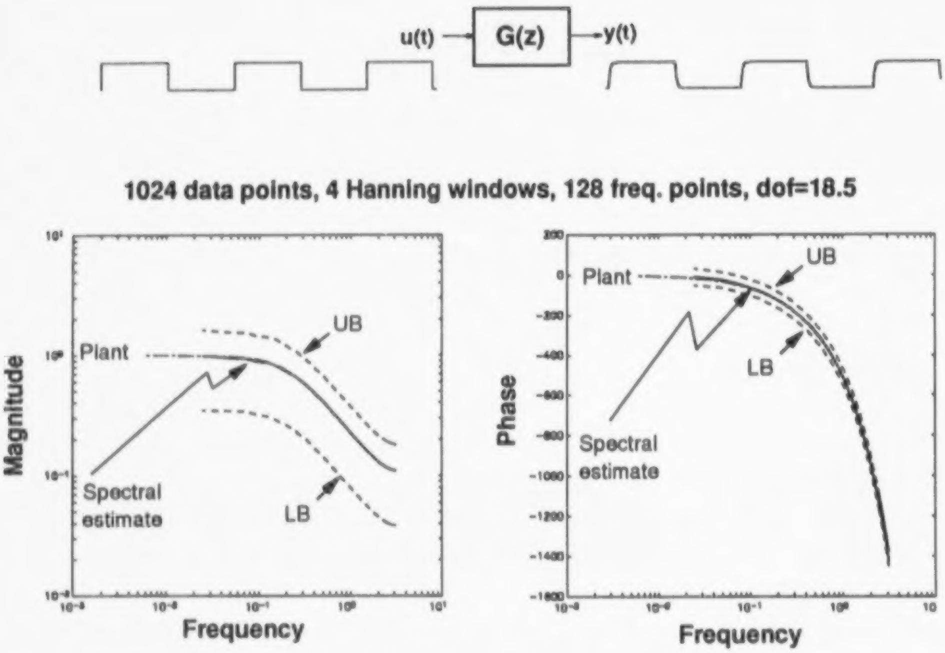


Figure 2.12: Top: Noise free process output for a square wave input. Bottom: Estimation of Bode plots and their bounds for the above case with 4 Hanning windows of length $M = 256$ each. Left: Magnitude spectrum. Right: Phase spectrum.

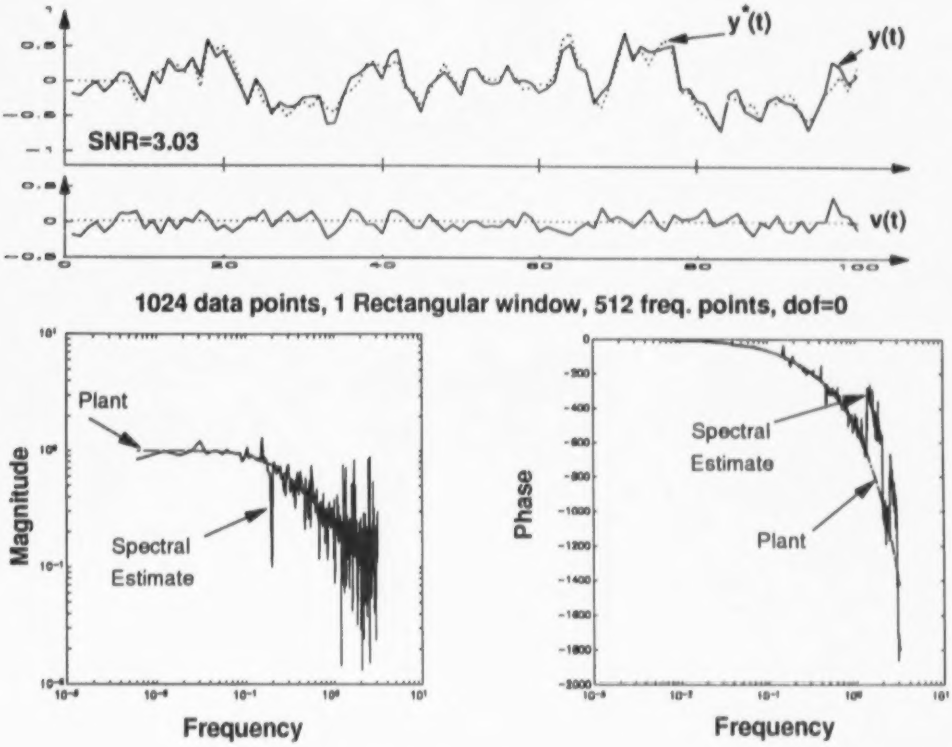


Figure 2.13: Top: Time series plots of noisy ($y(t)$) and noise free ($y^*(t)$) process outputs for a white noise excitation. $v(t)$ is the white measurement noise sequence (SNR=3.03). Bottom: Estimation of Bode plots for the above case with no spectral smoothing. Bottom-Left: Magnitude spectrum. Bottom-Right: Phase spectrum.

2.7.2 Case: Data corrupted by white measurement noise

In process industries, often the data record are said to be corrupted by white noise due to faults in the sensors and other measuring devices. This subsection therefore explores how signal processing methods perform in presence of white measurement noise. Figure 2.13 shows a section of the process output from Figure 2.5 corrupted by a significant level of white noise (SNR=3.03). The sequence of white noise is also shown in Figure 2.13. It is assumed that $u(t)$ and noise $v(t)$ are uncorrelated.

Case-1: The unsmoothened spectral estimates in Figure 2.13 shows significant variance and the phase spectrum shows bias at some frequencies.

Case-2: The example shown in Figure 2.14 is similar to the previous case except that a Hanning window is used to shape the data. This results in: (a) significant reduction in variance in the spectral estimates; but (b) no improvement in the quality of phase spectral estimates (on the contrary it deteriorates).

Case-3: However with the use of Welch's smoothing method the quality of the spectral

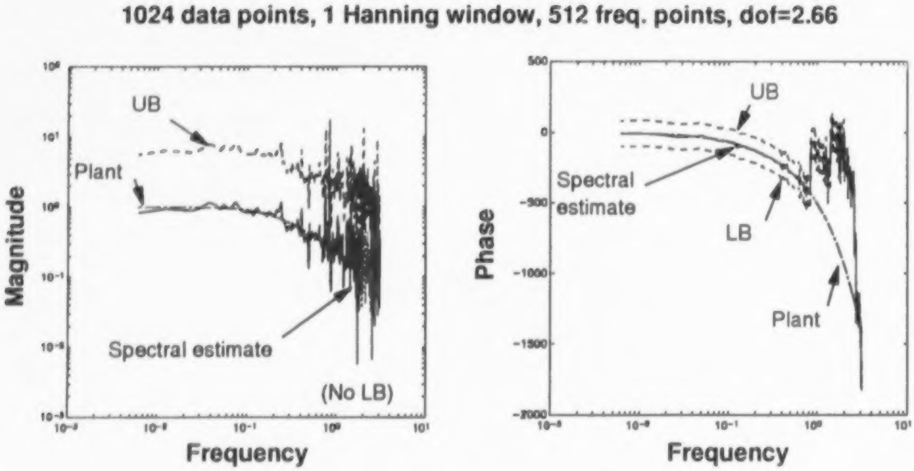


Figure 2.14: Estimation of Bode plots and their bounds for a plant corrupted by white noise (SNR=3.03) using (i) white noise excitation and (ii) 1 Hanning window of length $M = 1024$. Left: Magnitude spectrum. Right: Phase spectrum.

estimates improve as illustrated in Figure 2.15. The tighter confidence bounds on the estimated spectrum is due to a high value of dof which essentially stems from the use of large number of windows. This figure also shows the squared coherency spectrum, which is at a satisfactory level (*i.e.* closer to 1) for most frequencies except at the higher frequencies.

Case-4: The above case is repeated in Figure 2.16, except that Hanning windows are replaced by rectangular windows to check the effectiveness of Bartlett's smoothing (*i.e.* just data averaging without using any windows) in the presence of white noise. Expectedly the estimated spectrums in Figure 2.16 shows: (a) more variance; (b) more relaxed confidence bounds owing to lesser dof when compared with the earlier case; and (c) squared coherence values are lower compared to the previous case. Nevertheless this illustration conveys that:

- Data averaging is very essential and in fact it is more important than data windowing and this becomes obvious after comparing the phase spectrums in Figures 2.14, 2.15 and 2.16.
- Figure 2.15 shows higher squared cohrerency (owing to data windowing) than shown in Figure 2.16. Consequently the estimated spectrums in Figure 2.15 reflect the true Bode plots more accurately than in Figure 2.16.

In fact the squared coherency plot is very valuable because *it gives a measure of both quality*

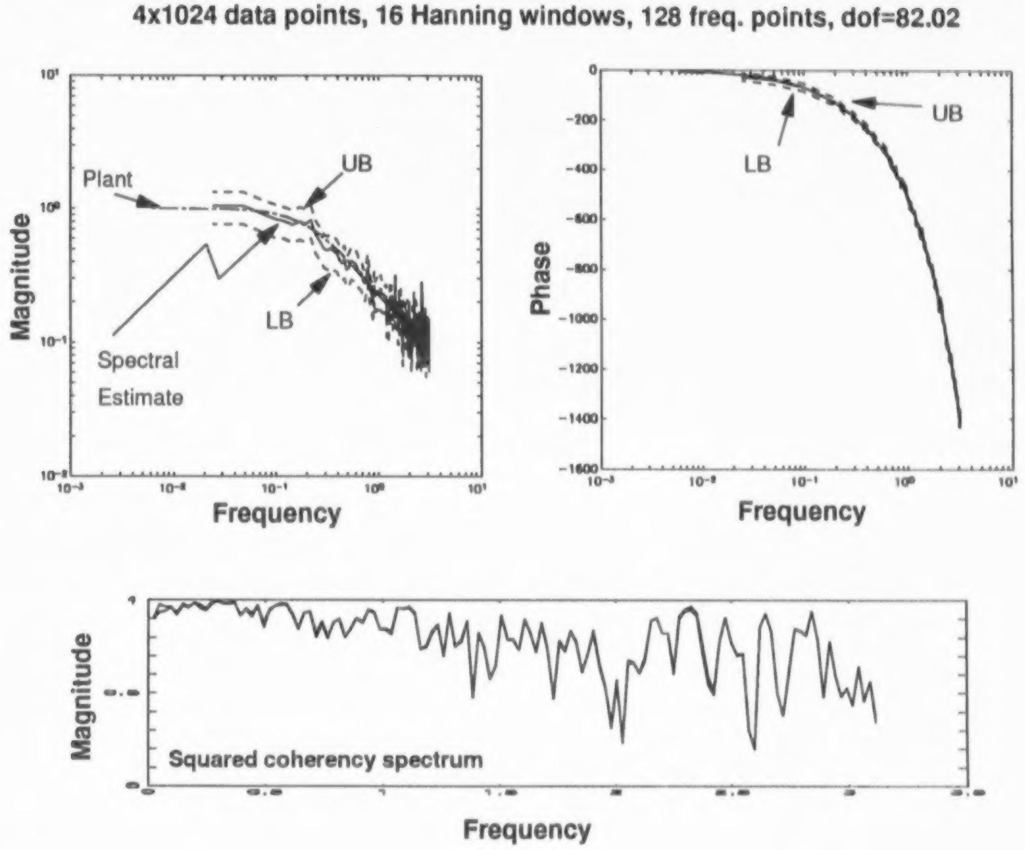


Figure 2.15: Estimation of Bode plots, their bounds and coherency spectrum for a plant corrupted by white noise (SNR=3.03) using (i) white noise excitation and (ii) 4 Hanning windows of length $M = 256$ each. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Squared coherency spectrum.

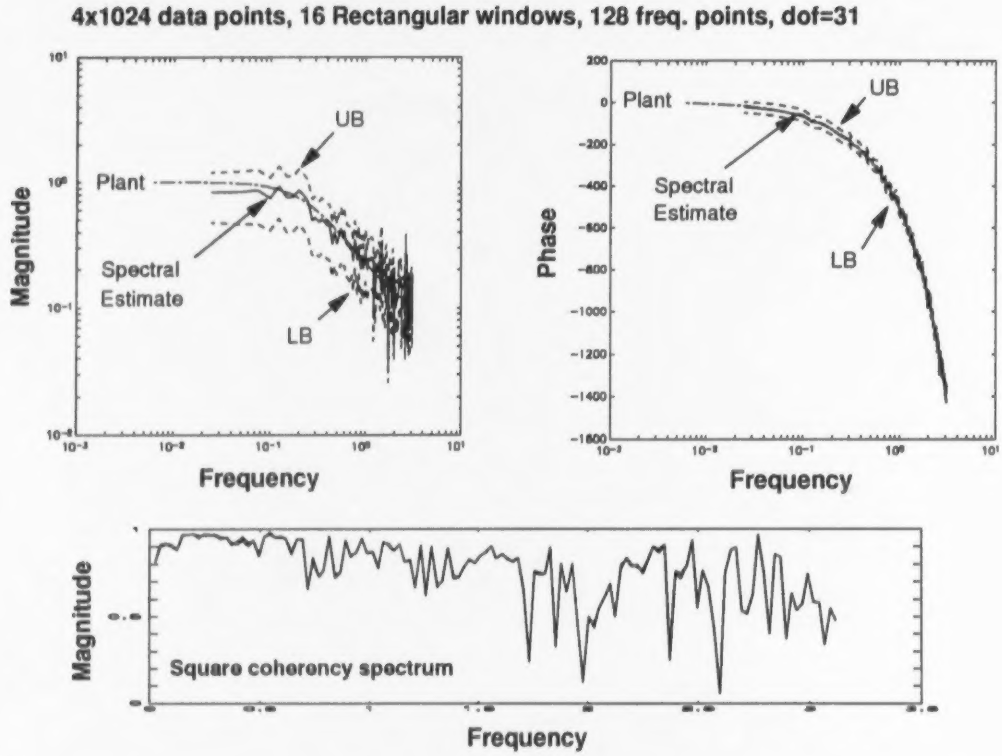

4x1024 data points, 16 Rectangular windows, 128 freq. points, dof=31

Figure 2.16: Estimation of Bode plots, their bounds and coherency spectrum for a plant corrupted by white noise (SNR=3.03) using (i) white noise excitation and (ii) Bartlett's smoothing method. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Squared coherency spectrum.

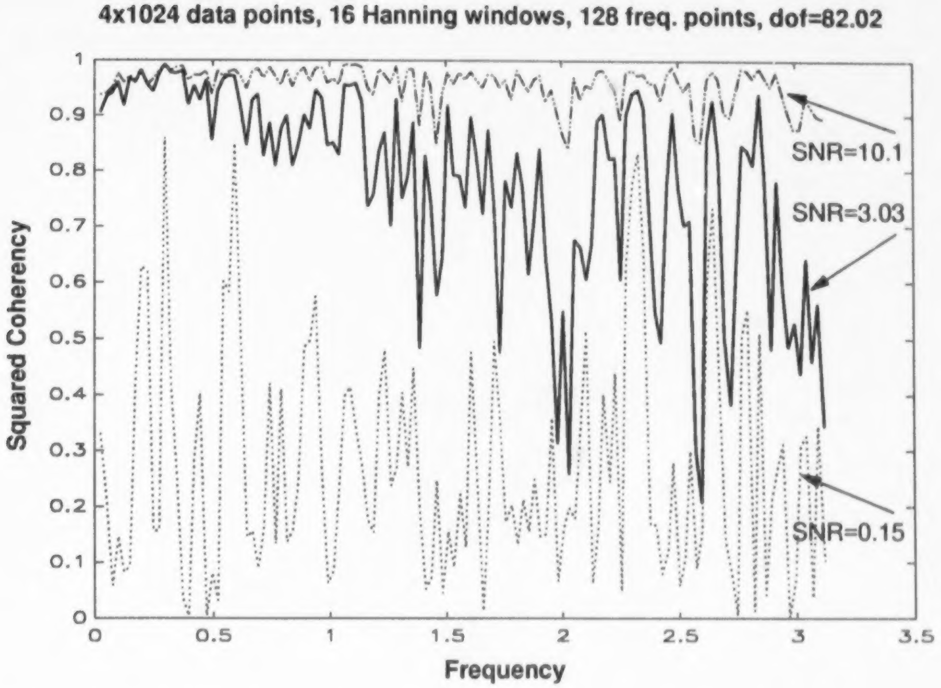


Figure 2.17: Effect of level of white measurement noise (SNR) on the squared coherency spectrum.

and reliability of the estimated spectrums. This is illustrated by an example in Figure 2.17 which shows higher coherency for richer signals and *vice-versa*, thus indicating *higher level of confidence and reliability on the estimated spectrums for higher values of the squared coherency.*

2.7.3 Case: Data corrupted by ARMA noise

Chemical processes are often prone to low frequency disturbances such as random walk and ARMA or colored-type noise disturbances. Low frequency disturbances tend to make the data record non-stationary and sometimes they may also exhibit nonlinear traits, thus posing challenges for both signal processing and system identification. The effectiveness of signal processing methods are evaluated in this section for output corrupted by ARMA noise. Cases of more difficult noise situations are presented in the next chapter.

The present case assumes that the output $y^*(t)$ in Figure 2.5 is corrupted by the ARMA noise model:

$$v(t) = \frac{1 + 0.5q^{-1} - 0.2q^{-2}}{1 - 0.999q^{-1}}\xi(t) \tag{2.71}$$

with SNR$\approx$ 3.0 (which is significant). Figure 2.18 shows both the noise free ($y^*(t)$) and

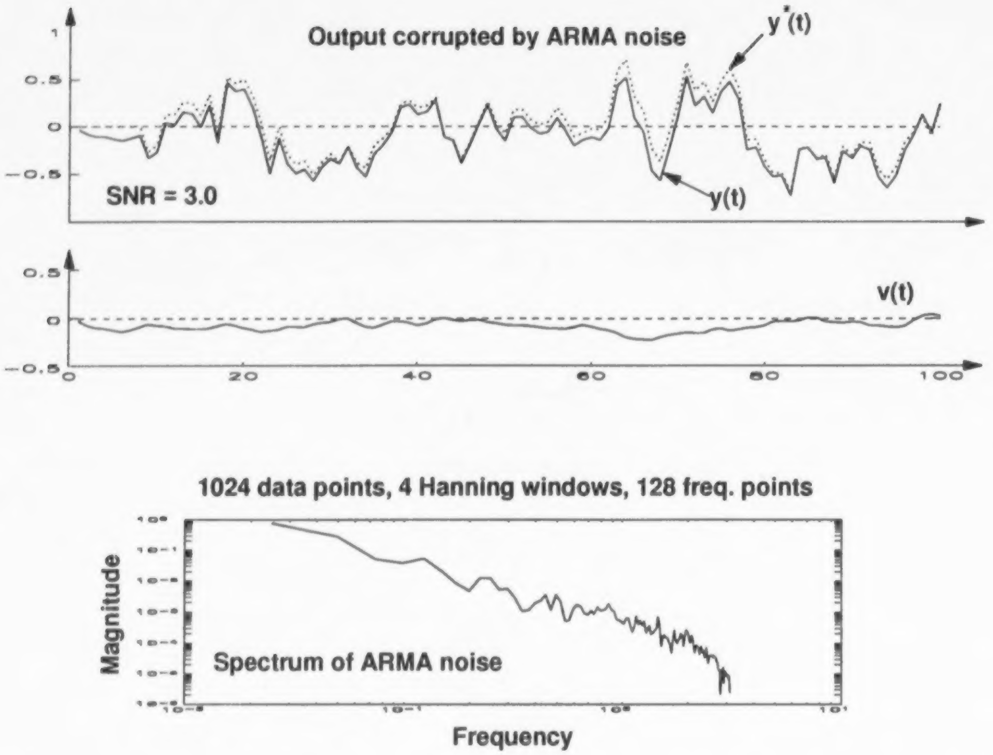


Figure 2.18: Top: Time series plots of noise free output ($y^*(t)$) and output corrupted by ARMA noise ($y(t)$) for white noise excitation. $v(t)$ is the ARMA noise sequence (SNR$\approx$ 3.0). Bottom: Magnitude spectrum of $v(t)$.

noisy ($y(t) = y^*(t) + v(t)$) process output together with a colored (ARMA) noise sequence $v(t)$. Magnitude spectrum of the ARMA noise in Figure 2.18 shows richness at the lower frequencies in contrast to the flat spectrum for a white noise signal.

The nonstationarity in $y(t)$ is first removed before applying signal processing methods ($y(t)$ is mildly nonstationary under the influence $v(t)$ whose nonstationarity is apparent in Figure 2.18). It should however be noted that nonstationarity in the data due to noise cannot always be removed completely and this often makes the job of identification or signal processing difficult. In the present case Figure 2.19 reveals biases in the estimated spectrums at the lower frequencies because the coherency in this region is small. Furthermore, smaller coherency at the lower frequencies is attributed to the significant level of noise in this frequency range.

One way to ameliorate the problems at lower frequencies (in this case) is to provide input excitation, $u(t)$, that is rich enough at these frequencies. A signal that is rich at the lower frequencies can be obtained by filtering the white noise input through a low pass filter. The time series and spectrum of the low pass filtered excitation $u^f(t)$ are depicted

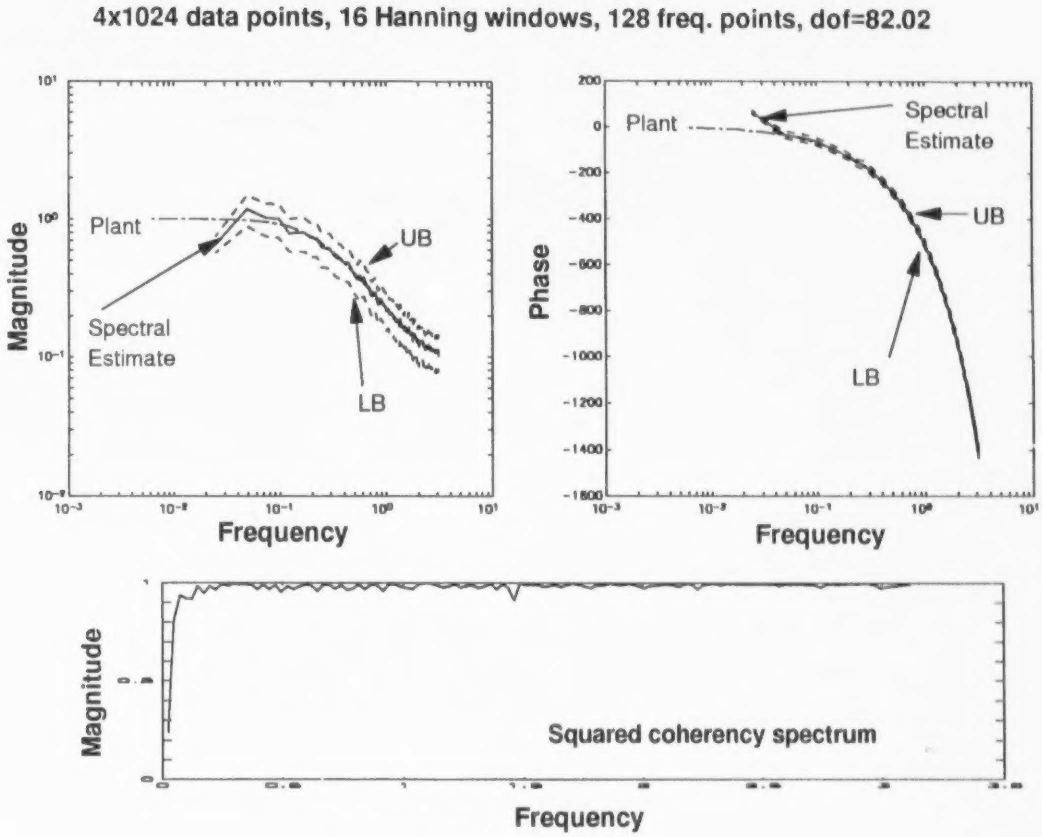


Figure 2.19: Estimates of Bode plots and coherency spectrum for a plant corrupted by ARMA noise (SNR$\approx$ 3.0) using (i) white noise excitation and (ii) 16 Hanning windows of length $M = 256$ each. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Square coherency spectrum.

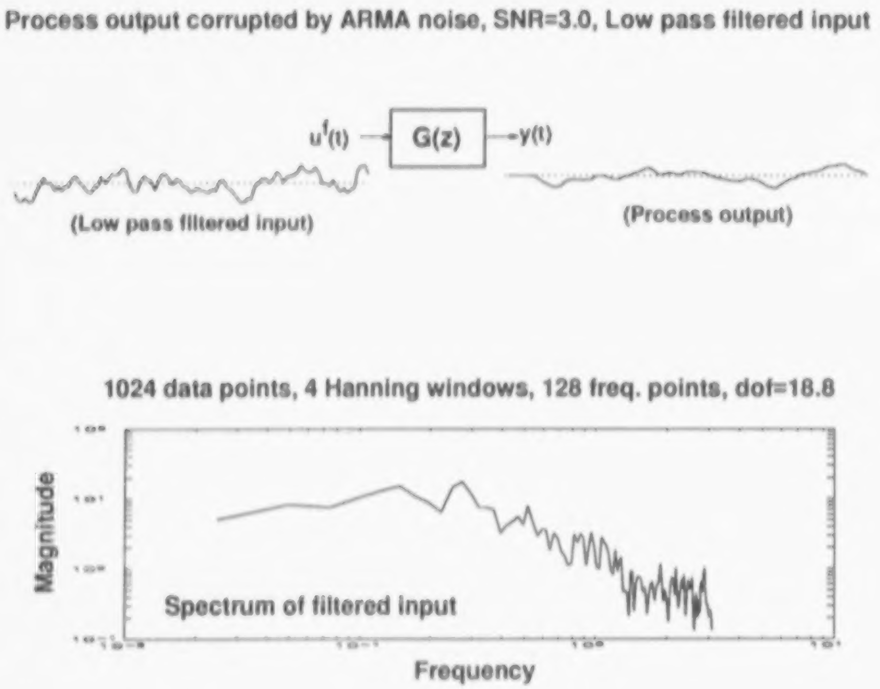


Figure 2.20: Top: Time series plots of low pass filtered white noise input $u^f(t)$ and its output $y(t)$ that is corrupted by ARMA noise (SNR$\approx$ 3.0). Bottom: Magnitude spectrum of filtered input.

in Figure 2.20 for an arbitrary filter $1/1 - 0.8q^{-1}$. The coherency plot in Figure 2.21 now shows richness at the lower frequencies as a result better estimates are obtained for these frequencies.

2.8 Estimation of model-plant mismatch (MPM)

The dynamics of a process can be mathematically expressed in several ways. But in a real situation (especially for chemical plants) the dynamics of a plant is are generally complicated, and therefore such mathematical models can only capture the most dominant dynamics. Hence in practice mathematical models are always associated with some uncertainty which is often called model plant mismatch (MPM).

For model based controllers the MPM acts as bounded disturbance which can adversely affect its stability and performance. But if information on MPM is available, then

---

[14] A version of the Section *'Estimation of model plant mismatch'* appeared in [6, 16].

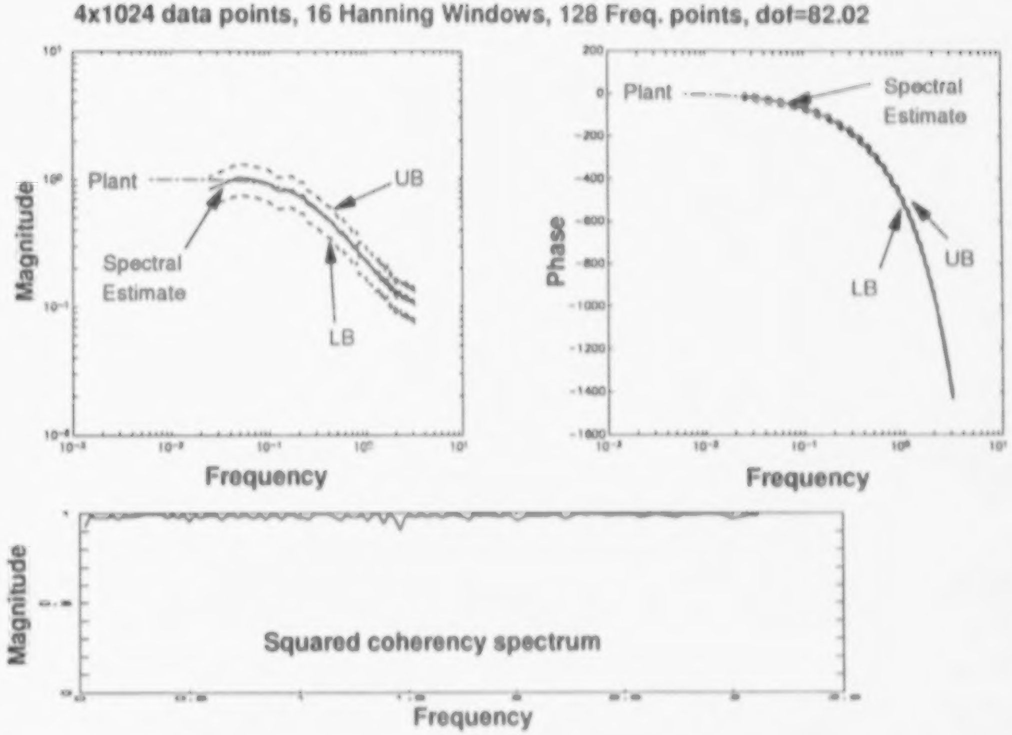

4x1024 data points, 16 Hanning Windows, 128 Freq. points, dof=82.02

Figure 2.21: Estimation of Bode plots and coherency spectrum for a plant corrupted by ARMA noise (SNR=3.03) using (i) low pass filtered white noise input and (ii) 16 Hanning windows of length $M = 256$ each. Top-left: Magnitude spectrum. Top-right: Phase spectrum. Bottom: Square coherency spectrum.

this knowledge can be used to design controllers that are robust to such modelling errors and these issues are further deliberated upon in Chapters 5 and 6.

It should be understood that the MPM is a non-stochastic term, but in practice it is always associated with measurement noise plus other disturbances which together is called the residual error $e(t)$ depicted in Figure 2.1. For model based controllers it therefore becomes important to extract MPM from $e(t)$, because it is the MPM that causes instability.

In reality the MPM is never known and it can best be represented nonparametrically such as in the frequency domain. If the MPM is designated by $\tilde{G}(\omega)$, then from Figure 2.1 it is obvious that:

$$\tilde{G}(\omega) = G(\omega) - \hat{G}(\omega) \tag{2.72}$$

If the plant is noise free *i.e.* $v(t) = 0$, equations-(2.34), (2.41), (2.42) and (2.72) can be used to obtain the following expression for MPM [17, 18, 19]:

$$\tilde{G}_M(\omega) = \frac{\Phi_{eu}(\omega)}{\Phi_{yy}(\omega)} \tag{2.73}$$

or

$$|\tilde{G}_M(\omega)| = \sqrt{\frac{\Phi_{ee}(\omega)}{\Phi_{yy}(\omega)}} \tag{2.74}$$

The subscript M in equations-(2.73) or (2.74) designates that it is a spectral estimate of $\tilde{G}(\omega)$ (note: M signifies the number of frequency points for DFT computation, therefore the use of M is appropriate in this context *i.e.* in equation-(2.74)).

MPM thus represented by equations-(2.73) or (2.74) is nonparametric and it encompasses all information about the mismatches in process gain, time-constant and delay. The MPM in equations-(2.73) or (2.74) is therefore unstructured and its use in control design will ensure robustness to all structural uncertainties such as the mismatches in process gain, delay or time constant as discussed in Chapters 5 and 6.

The equations (2.73) or (2.74) are the theoretical estimates based on the asymptotic results. In practice however the given data are finite, therefore: (a) the spectral smoothing methods discussed in Section 2.5.8 are applied to obtain consistent and reliable estimates of $\tilde{G}_M(\omega)$; and (b) the confidence bounds are provided for $\tilde{G}_M(\omega)$ to account for the uncertainties in the estimates. For robust controller design only the upper confidence bound is required as it signifies maximum modelling error with a certain degree of confidence.

In presence of $v(t)$ the estimated $\tilde{G}_M(\omega)$ will be biased as illustrated by several examples in Section 2.7. If the biased $\tilde{G}_M(\omega)$ is denoted by $\tilde{G}_M^B(\omega)$, (superscript B indicates bias), then:

$$|\tilde{G}_M^B(\omega)| = \sqrt{|\tilde{G}_M(\omega)|^2 + \frac{\Phi_{vv}(\omega)}{\Phi_{uu}(\omega)}} \tag{2.75}$$

The above expression shows that in presence of $v(t)$:,

- A bias proportional to $\Phi_{vv}(\omega)/\Phi_{uu}(\omega)$ is added to $|\tilde{G}_M(\omega)|$.
- Bias is affected by the shape of $\Phi_{uu}(\omega)$. For example, the effect of roll off in $\Phi_{uu}(\omega)$ at the higher frequencies would be to amplify the noise and therefore add more bias in the estimated $|\tilde{G}_M(\omega)|$ at those frequencies.
- For colored noise *i.e.* $v(t) = H(q^{-1})\xi(t)$ $[\xi(t) \in \mathcal{N}(0, \sigma^2)]$ the biased estimate of MPM is:

$$|\tilde{G}^B_M(\omega)| = \sqrt{|\tilde{G}_M(\omega)|^2 + |H(\omega)|^2 \frac{\Phi_{vv}(\omega)}{\Phi_{uu}(\omega)}} \tag{2.76}$$

which shows that the bias is influenced by $H(q^{-1})$ in addition to $\Phi_{uu}(\omega)$. Therefore depending on the nature of $H(q^{-1})$, appropriate pre-filtering is required to minimize the effect of bias on $\tilde{G}_M(\omega)$.

In model based control, the problems of stability manifests from $\tilde{G}(\omega)$ rather than from noise or disturbance $v(t)$ (assuming the designed controller is stable). Therefore it makes sense only to estimate the $\tilde{G}(\omega)$ component and not the spectrum of $v(t)$ in the interest of a less conservative controller design. These issues are pursued further in Chapters 5 and 6. This section is only devoted to the methodologies (via signal processing) by which a good estimate of $\tilde{G}_M(\omega)$ can be obtained. In essence the central message is - *MPM bounds should be determined as accurately as possible because an inadequate estimation can lead to instability whereas an overly conservative bound can affect the performance.* Further, equations-2.75 and 2.76 suggests that $v(t)$ tends to make the estimated bounds more conservative.

To the best of author's knowledge, most research papers *e.g.* [17, 18, 19] tend to ignore the effect of bias in the spectral estimation of MPM. Yet the effect of bias can be very significant, because, the SNR is usually poor in $e(t)$. Often for good models (which is desirable) $||e|| < ||y||$ and it is obvious from equations-(2.1) and (2.72) that any noise in $y(t)$ gets transferred to $e(t)$. Therefore the bias term $\Phi_{vv}(\omega)/\Phi_{uu}(\omega)$ can usually be significant in the estimation of MPM. For example, in an extreme situation, when $\tilde{G}(q^{-1}) = 0$, $e(t)$ is simply $v(t)$ or $|\tilde{G}_M(\omega)| = \sqrt{\Phi_{vv}(\omega)/\Phi_{uu}(\omega)}$.

2.8.1 Spectral estimation of MPM - simulation example

In reality the true model-plant uncertainty is never known exactly and hence the effectiveness of the DFT method to estimate MPM can only be ascertained through contrived examples such as the following. Consider a third order overdamped plant: $G_2(s) = 1/(s+1)(3s+1)(5s+1)$, whose discrete equivalent for a sample time of $T_s = 1$ is:

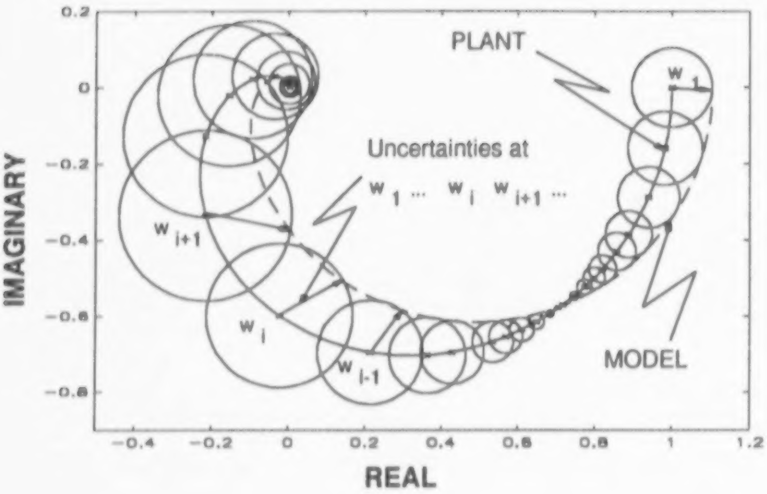


Figure 2.22: Nyquist plots of the plant, model and the uncertainty regions.

$$G_2(q^{-1}) = \frac{0.0077q^{-1} + 0.0212q^{-2} + 0.0036q^{-3}}{1 - 1.9031q^{-1} + 1.1514q^{-2} - 0.2158q^{-3}} \tag{2.77}$$

The following first order model with two numerator parameters was identified using least square technique by providing a square wave excitation:

$$\hat{G}_2(q^{-1}) = \frac{0.0419q^{-1} + 0.0719q^{-2}}{1 - 0.8969q^{-1}} \tag{2.78}$$

The Nyquist plot of the plant, model and the uncertainty regions are shown in Figure 2.22

The objective is to estimate the MPM for the system equations-(2.77) and (2.78) using signal processing methods discussed in Section 2.5. The use of various smoothing techniques to improve spectral estimates of Bode plots have already been illustrated via ample number of case studies in Section 2.7. The results of case studies in Section 2.7 are extended here to show how they can be applied to estimate the MPM spectrum [6, 16].

The effect of two extreme cases of input types (*i.e.* square wave and white noise) on the estimation of MPM are discussed in the presence and absence of measurement noise. The noise-free case is discussed first to examine the effect of input type on the spectral estimation of the MPM. The illustration on the left in Figure 2.23 and 2.24 show that

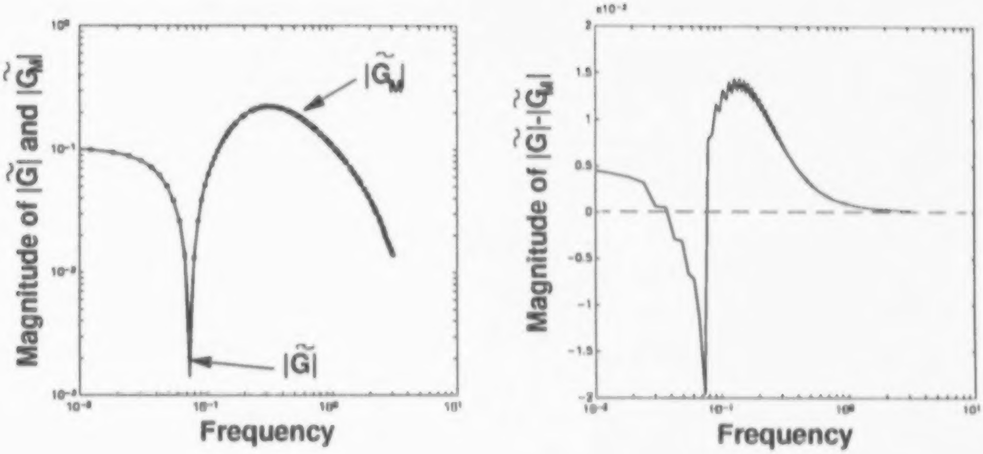


Figure 2.23: Left: Spectral estimation of MPM with square wave input for a noise free plant. Right: Error between true and estimated MPM.

$|\tilde{G}(\omega)|$ and $|\tilde{G}_M(\omega)|$ compare well when: (a) subjected to square type and white noise excitation ($N = 1024$); and (b) spectral smoothing methods are employed.

The plot on the right side in Figure 2.23 and 2.24 show baises between $|\tilde{G}(\omega)|$ and $|\tilde{G}_M(\omega)|$ for square and white noise inputs. These figures show lower biases for low frequency square wave input when compared with white noise excitation. But the bias corresponding to square wave input shows a pattern whereas for white noise it is randomly distributed about 0. The distribution of bias in Figure 2.23 confirms the fact that bias in the estimated spectrum is greater wherever the first and second derivatives of the transfer function w.r.t. ω are significant [15]. This fact shows up for square type input but for white noise excitation this is not perceptible.

For SNR$\approx$ 1.4, Figure 2.25 shows that the estimated bias in MPM is greater at higher frequencies for the square wave input in comparison with the white noise input. This result is contrary to the previous results for noise-free case. This is expected, because for a square input, $\Phi_{uu}(\omega)$ rolls off at the higher frequencies which magnifies the effect of noise at these frequencies according to equation-(2.75). Similarly, the flat spectrum due to white noise introduces lesser bias in the estimated MPM in presence of measurement noise at higher frequencies. Nevertheless the square input gives a better estimation at the lower frequencies because of more signal energies at these frequencies (note that: (a) usual spectral smoothing methods are employed for these cases; and (b) the level of noise is very

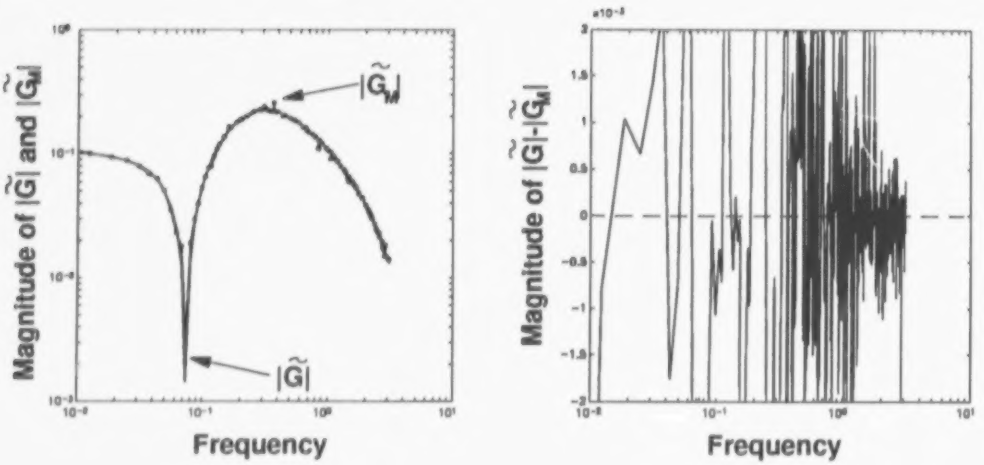


Figure 2.24: Left: Spectral estimation of MPM with white noise input for a noise free plant. Right: Error between true and estimated MPM.

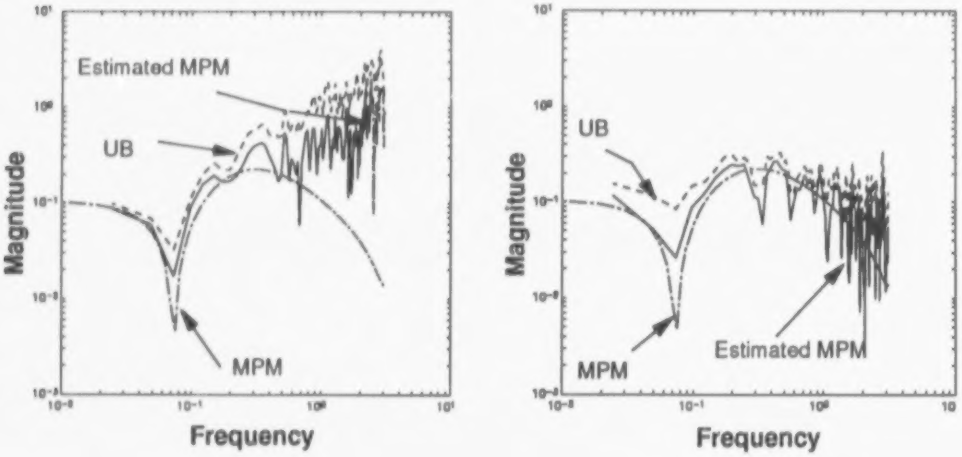


Figure 2.25: Spectral estimation of MPM in the presence of white measurement noise (SNR$\approx$ 1.4). Left: with square wave input, Right: with white noise input.

significant).

These results indicate that a square type input with proper smoothing methods can give a good and reliable estimate of the MPM from $e(t, N)$ provided that the level of measurement noise is sufficiently low. In the presence of measurement noise, a white noise excitation will give a better estimate of MPM for most frequencies and especially at the higher end of the spectrum. However if accuracy in the estimation of MPM is desired at the lower frequencies, then a square type input is also required. The coherency plots discussed in Section 2.7 must also be referred to gain confidence in the estimated spectrum. In this particular illustration *i.e.* left figure in Figure 2.25, the coherency spectrum $\kappa^2_{eu}(\omega)$ is relatively large (close to 1) at the lower frequencies due to square wave excitation. This results in a good estimate of the upperbound in comparison to the white noise excitation at these lower frequencies.

2.9 Conclusions

The main contributions of this chapter are:

- The field of signal processing is introduced with an interesting historical perspective. Different signal processing results that are relevant to system identification are brought together. Importance of these results are highlighted via example dependent case studies and illustrations.
- This chapter compactly gives the gist of signal processing methods with a tutorial flavor such that they can be easily adopted by the chemical engineering practitioners.
- The significance of signal processing methods presented in this chapter are summarized in the following:
    - $\gamma_{xx}(k)$ gives the nature of correlation for $x(t, N \to \infty)$ whereas $\gamma_{yu}(k)$ determines the correlation between $y(t, N \to \infty)$ and $u(t, N \to \infty)$. $\rho_{xx}(k)$ and $\rho_{yu}(k)$ are the normalized versions of $\gamma_{xx}(k)$ and $\gamma_{yu}(k)$ respectively. $\hat{\rho}_{xx}(k)$ corresponds to finite N and it is assumed $\rho_{xx}(k) - \hat{\rho}_{xx}(k) \in \mathcal{N}(0, \sigma^2)$ and the similar argument holds good for $\hat{\rho}_{yu}(k)$.
    - The acf helps to check if $x(t, N) \in \mathcal{N}(0, \sigma^2)$ or $x(t, N) \in$ (MA(r) or AR(p)). AR(p) is ascertained by examining the pacf. Similarly ccf provides information on the direction of gain, delay and impulse response.
    - Periodic and non-periodic time domain signals are converted into frequency domain by using Fourier series and FT respectively. Periodic signals yield discrete frequencies whereas nonperiodic signals give the effect of continuous frequency. FT$[x_c(t, T)]$ is related to Laplace transform whereas FT of ideally sampled non-periodic $x_c(t, T)$ *i.e.* FT$[x(t, N)]$ is same as the $z-$transform.

- DFT is applicable to periodic $x(t, N)$ and its numerically faster (or practical) version is called FFT.
- $\Phi_{xx}(\omega)$ describes the energy distribution of $x(t, N)$ for different $\omega \in [0, \omega_N]$. Similarly $\Phi_{yu}(\omega)$ describes relation between $y(t, N)$ and $u(t, N)$ for $\omega \in [0, \omega_N]$. The ratio between $\Phi_{yu}(\omega)$ and $\Phi_{uu}(\omega)$ is used to estimate the Bode plots.
- It is essential to smoothen the periodograms to obtain a good estimate of the signal spectrum. Averaging of segmented signals and data windowing are two principal smoothing techniques. A combination of these two methods is known as Welch's smoothing which usually leads to better estimation of the spectrum.
- Statistical methods such as χ^2 and Fisher's F distibution are used to derive confidence bounds for the estimated spectrum. The confidence bounds are tighter if more data segments are considered and it also depends on the choice of windows.
- $\bar{\kappa}^2_{yu}(\omega)$ gives a measure of the reliability and quality of the estimated Bode plots. $\bar{\kappa}^2_{yu}(\omega)$ varies between 1 and 0, signifying infinite SNR (*i.e.* very reliable estimates) to SNR=0 *i.e.* unreliable spectral estimates.
- For a noise free plant, low frequency inputs such as square wave signals give no spectral variance whereas white noise excitation always give variance. However, for noise corrupted output low frequency signals give biased spectral estimates at the higher frequency end.
- The spectral estimation techniques discussed above are also extended to estimate the MPM.

Notation

Roman Characters

a	Coefficients of AR model
c	Coefficients of MA model
e, E	Time and frequency domain error (or model-plant mismatch) signal
$\mathcal{E}$	Expectation operator
f, F	Fisher distribution value, Fisher's distribution
G	Transfer function
h	Impulse Response coefficients (*e.g.* h_k is k^{th} impulse response coefficient)
j	Complex number $\sqrt{-1}$
M	Number of frequency domain points or length of a data window
N	Number of data points for a time series data
$\mathcal{N}$	Normal distribution
$\mathcal{P}$	Probablity
q	Forward shift operator $(i.e. x(q^k) = x(t+k))$
q^{-1}	Backward shift operator $(i.e. x(q^{-k}) = x(t-k)\ )$
r	Order of AR model
s	Laplace domain *e.g.* $G(s)$
t	Sampling time instant
T	Time period of a signal
T_s	Sampling time
u, U	Time and frequency domain input signal
w	Window function
W	DFT operator $(W = e^{-j\Delta\omega T_s})$
y, Y	Time and frequency domain plant output signal
z	Discrete domain *e.g.* $G(z)$
Z	White noise in frequency domain

Greek Characters

α	Confidence limit
γ	Covariance function
$\gamma_{xx}(k)$	Auto covariance function of $x(t, N)$ at lag k
$\gamma_{yu}(k)$	Cross covariance function between $y(t, N)$ and $u(t + k, N)$
δ	Unit impulse *e.g.* $\delta(n) = 1$ implies at time n only, elsewhere 0
κ	Coherence
μ	Population mean (*i.e.* $\mu = \mathcal{E}(x(t, N)))$
ν	Degree of freedom
ξ	White noise sequence
$\rho(k)$	Correlation at lag k
$\rho_{xx}(k)$	Auto correlation of $x(t, N)$ at lag k
$\rho_{yu}(k)$	Cross correlation between $y(t, N)$ and $u(t + k, N)$
σ	Standard Deviation (or σ^2 is the variance)
υ	Time and frequency domain process noise signal (unmeasurable)
$\phi(k)$	Partial autocorrelation function at lag k
Φ	Power spectrum or simply spectrum
$\Phi_{xx}(\omega)$	Auto spectrum of $x(t, N)$ at frequency ω
$\Phi_{uy}(\omega)$	Cross spectrum between $u(t, N)$ and $y(t, N)$
φ	Phase spectrum for transfer function
φ_{yu}	Phase spectrum for cross spectrum
χ^2	Chi square distribution
ω	Frequency ($\omega = 2\pi/T_s$)
ω_N	Nyquist Frequency ($\omega_N = \omega/2$)

Superscripts, subscripts and symbols

c	Continuous time signal (*e.g.* $x_c(t)$)
H	Complex conjugate (*e.g.* X^H)
i	Imaginary component (*e.g.* X_i)
L	Lower bound (*e.g.* X^L)
r	Real component (*e.g.* X_r)
U	Upper bound (*e.g.* X^U)
$\bar{}$	Sample mean or a smoothened value (*e.g.* $\bar{x}$)
$\hat{}$	Estimated value or a property based on finite data record (*e.g.* $\hat{X}$)
$*$	True value of a signal (*e.g.* $x^*(t)$)
$\star$	Convolution
$\tilde{}$	Difference between true and its estimated value (*e.g.* $\tilde{x}(t)$)
$\|\| \bullet \|\|$	Norm (*e.g.* $\|\|X\|\|$)
$\| \bullet \|$	Magnitude (*e.g.* $\|X\|$)
$\angle$	Angle (*e.g.* $\angle X$)

Abbreviations

acf	Autocorrelation function
AR	Autoregressive
ccf	Cross correlation function
DFT	Discrete Fourier transform
FFT	Fast Fourier transform
FT	Fourier transform
LHS	Left hand side
MA	Moving average
MPM	Model-plant mismatch
pacf	Partial autocorrelation function
pdf	Probability distribution function
RHS	Right hand side
Var	Variance

Bibliography

[1] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing.* Prentice Hall, New Jersey, USA, 1975.

[2] R.N. Bracewell. "The Fourier Transform". *Scientific American*, pages 86 – 95, June 1989.

[3] E.A. Robinson. "A Historical Perspective of Spectrum Estimation". *Proceedings of the IEEE*, 70(9):885 – 907, 1982.

[4] A.V. Oppenheim, A.S. Willsky, and I.T. Young. *Signals and Systems.* Prentice Hall of India Pvt. Ltd, New Delhi, India, 1987.

[5] D.R. Brillinger. *Time Series Data Analysis and Theory.* Holt, Rinehart and Winston Inc., New York, USA, 1975.

[6] P. Banerjee and S.L. Shah. "The Role of Signal Processing Methods in the Robust Design of Predictive Control". *Automatica*, 31(5):681 – 695, 1995.

[7] B. Huang and S.L. Shah. "On-line control performance monitoring of MIMO processes". In *American Control Conference*, pages 1250 – 1254, Seattle, WA, USA, 1995.

[8] L.R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing.* Prentice Hall of India Pvt. Ltd., New Delhi, India, 1990.

[9] J.D. Cryer. *Time Series Analysis.* PWS-KENT Publishing Company, Boston, USA, 1986.

[10] G.M. Jenkins and D.G. Watts. *Spectral Analysis and its Application.* Holden-Day, San Franscisco, USA, 1968.

[11] Kay. S.M. and S.L. Marple. "Spectrum Analysis - A Modern Perspective". *Proceedings of the IEEE*, 69(11):1380 – 1418, 1981.

[12] W.D. Stanley, G.R. Dougherty, and R. Dougherty. *Digital Signal Processing.* Prentice-Hall, New Jersey, USA, 1984. Second edition.

[13] M.B. Priestley. *Spectral Analysis and Time Series, Vol-1 and 2.* Academic Press, London, UK, 1981.

[14] R.K. Pearson and B.A. Ogunnaike. "Application of Coherence Analysis in Linear and Nonlinear Input/Output Model Identification". In *AIChE Annual Meeting, San Francisco.*, 1994.

[15] L. Ljung. *System Identification-Theory For The User.* Prentice Hall, New Jersey, USA, 1987.

[16] P. Banerjee and S.L. Shah. "Estimation of Model-Plant Uncertainty and its role in Robust Design of Predictive Control". In *12th IFAC World Congress*, volume 2, pages 321 – 326, Sydney, Australia, July 18-23, 1993.

[17] R.L. Kosut. "Adaptive uncertainty modelling: on-line robust control design". In *American Control Conference*, pages 245–250, Minneapolis, USA, 1987.

[18] Y.C. Zhu. "Estimation of transfer functions: asymptotic theory and a bound of model uncertainty". *Int. J. Control*, 49:2241 – 2258, 1989.

[19] D.E. Rivera, J.F. Pollard, and C.E. Garcia. "Control-Relevant Prefiltering: A Systematic Design Approach and Case Study". *IEEE Trans. on AC*, 37(7):964 – 974, 1992.

Chapter 3

System Identification - Case Studies

Different system identification methods are reviewed and their application is illustrated via two industrial case studies.

3.1 Introduction

System identification (ID) is the subject of mathematically describing the dynamic behavior of a system based on experimental data [1]. Hence for system ID there is need for: (a) experimental data, (b) an appropriate mathematical model structure and (c) a method by which such a model can be estimated. The choice of these prerequisites for system ID has a significant effect on the final outcome *i.e.* the identified model and is largely influenced by the intended application.

System ID is used in several engineering applications and is also applied in areas such as biology, environmental science and econometrics [1]. Some engineering applications of system ID include signal processing, fault detection, pattern recognition and process control [1]. The main emphasis of this chapter and also in this thesis is to use system ID in such a way that the resultant model can be readily used in the design and/or implementation of process control.

The dynamics of a ***physical system*** (also referred as a ***plant*** or ***process***) can be modelled in several ways ranging from simple empirical to detailed mechanistic models. Familiarity with the process and its physical laws are imperative in building ***ab initio*** mechanistic models which are usually a combination of algebraic and integro-differential/partial differential equations. Many times such mechanistic models can successfully describe the system behavior however: (a) such models are generally complex and tedious to develop, (b) on most occasions it is difficult to design process controllers that are based on such detailed mechanistic models and (c) ***frequently the real plant behavior (particularly for a chemical process) is too complicated to be emulated even by such mechanistic models.***

---

[1] A version of this chapter has been published as: P.Banerjee, S.L.Shah, S.Niu and D.G.Fisher, 'Identification of dynamic models for the Shell benchmark problem', ***Journal of Process Control***, vol 5, No. 2, pp. 85-97, 1995.

Therefore empirical or semi-empirical models are more popular in system ID. Empirical models are developed without any ***a priori*** knowledge of the physical system and hence such models are also called ***black-box*** models. Similarly models that are based on some minimal ***a priori*** information about the plant are called semi-empirical or ***grey-box*** models. The advantages of empirical or semi-empirical models are: (a) they are simple; (b) often detailed insight about the plant is not required; (c) design of experiments to estimate such models is simple; and (d) ***it is relatively easy to design controllers that are based on such empirical/semi-empirical models.*** Black/grey-box models are often sufficient for many applications such as in process control; where it is required that the models adequately describe the important and dominant dynamics of the process only around the current operating point. Models of lower complexity are also desirable for process control because they yield simple controller designs. Therefore ***system identification generally focuses on estimating simple parsimonious models that adequately describe the dominant dynamics of the plant.*** The emphasis of this chapter is therefore on identifying black/grey-box models that can eventually be used for controller design. In Chapters 5 and 6 it is shown how estimated black/grey-box models can be used for the design of robust model predictive controllers.

Different empirical/semi-empirical models used to represent the plant dynamics are well known in the literature [1, 2, 3, 4]. A partial list of such models that are relevant to this thesis are summarized in Section 3.2. The characteristics of these models and their estimation methods are described in Sections 3.3 to 3.7. In particular, Section 3.3 summarizes estimation of frequency response models. Estimation of step and finite impulse response (FIR) models is described in Section 3.4. Approximation of model dynamics using different types of input-output (I/O) transfer function models such as ARX models and their estimation procedures are discussed in Section 3.5. Section 3.6 briefly outlines estimation of orthonormal models. A more detailed discussion on the estimation of orthonormal models is presented in Chapter 4. State space models are briefly mentioned in Section 3.7.

The confidence of the estimated parameters is usually defined by statistical bounds and this is discussed in Section 3.8. The quality of the estimated model is ascertained using time and frequency domain model validation methods which are summarized in Section 3.9.

System ID is a well established field with a significant amount of published results [1, 2, 5]. System ID methods are regularly used by many process industries and consultancy/engineering firms[2] for their controller design. Yet, not many published results are available that illustrate the use of system ID methods in a systematic manner via real industrial applications. Therefore a major portion of this chapter is dedicated to systematic application of system ID methods. They are illustrated by applying them to two industrial case studies.

---

[2] Companies such as Setpoint, DMC, Gensym, Honeywell-Profimatics etc. extensively use system ID

The plant for the system ID case study in Section 3.10 is in the form of a computer program that very closely emulates the behavior of an industrial distillation column. This problem is also referred as the Shell benchmark problem because Shell Canada[3] was primarily instrumental in formulating this identification exercise. In order to evaluate different system ID methods, the *Systems and Control Division* of *Canadian Society for Chemical Enginering* made this Shell benchmark problem for identification available to groups[4] across Canada. The outcome of this identification exercise was summarized at the 1992 *Canadian Chemical Engineering Conference* (CSChE-92), Toronto, Canada [6]. The Shell problem is included in this thesis because: (a) University of Alberta (U of A) was one of the participants; the author's contribution was significant [7]; and (b) it serves as an excellent example to illustrate the application of system ID techniques outlined in this chapter.

The second case study presented in Section 3.11 deals with identifying models from batches of industrial plant I/O data. The process in this case is a catalytic reactor in a fertilizer plant that converts NH_3 to NO_2 for Urea production.

Identification of both these industrial case studies is the main contribution of this chapter because these case studies serve as excellent examples on how to use system ID methods in a systematic way. The identification case studies are followed by concluding remarks in Section 3.12.

3.2 Modelling linear systems

The dynamic model is estimated as indicated by the block diagram in Figure 2.1. The blocks $G(z)$ and $\hat{G}(z)$ in that figure denote the ***discrete linear time invariant*** plant and its model respectively. This system is mathematically described as:

$$\begin{aligned} y(t) &= \underbrace{G(q^{-1})u(t)}_{y^*(t)} + v(t) \\ \hat{y}(t) &= \hat{G}(q^{-1})u(t) \end{aligned} \tag{3.1}$$

Remark-2.1 in Chapter 2 also applies here but in addition the following is to be noted:

---

methods for controller design.

[3] The Shell benchmark problem was made available by Dr. Barry Cott of 'Process Control Group', Shell Canada Products Limited, Sarnia, Ontario, Canada [6].

[4] (a) Participating Universities: U of Alberta, U of Toronto and U of Western Ontario. (b) Participating Companies: Shell Canada, Sarnia and Sunoco Incorporated, Sarnia.

Note 3.1 *Although the operator q^{-1} is used for time domain manipulations, for simplicity reasons q is used as an argument for the polynomials instead of q^{-1}. As an illustration, the polynomial $B(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \cdots + b_{nb}q^{-nb}$ would be denoted by $B(q) = b_1q^{-1} + b_2q^{-2} + \cdots + b_{nb}q^{-nb}$ or $G(q^{-1})$ in equation-(3.1) by $G(q)$.*

The plant $G(q)$ can be modelled in several ways and further, these models can have several classifications. It is also possible in most cases to convert one form of model to another, but the choice of these models largely depends on: (a) the intended application (*e.g.* controller design); (b) parsimony in parameters; and (c) method used to estimate or identify these models. The model $\hat{G}(q)$ addressed in this thesis can be classified as:

- Parametric or non-parametric model.
- Structured or unstructured model.
- Equation error (EE) or output error (OE) model.
- Input-output (I/O) or state space (SS) model.
- Black box or grey box model.

Dynamic models referred in this thesis fall in more than one of the above mentioned classes and they can be represented as:

- Frequency response models.
- Step response and finite impulse response (FIR) models.
- Transfer function models.
- Orthonormal function models.
- State space models.

A brief description of the models, their estimation and validation methods, parameter confidence bounds and their use in modelling industrial systems are presented in the ensuing sections.

3.3 Frequency response models

Frequency response models are popularly known as Bode and Nyquist "models" and need no introduction. Frequency response models are ***unstructured*** and ***non-parametric*** as they have no formal mathematical structure; instead they are represented by a set of frequency dependent complex numbers or equivalently by a set of magnitudes and phases.

Frequency response models are used to analyze the stability of closed loop control systems. In Chapters 5 and 6 it is shown how frequency response models are used to design robust controllers. Frequency response models are also used to validate mathematical models as described later in this chapter.

3.3.1 Estimation of frequency response models

There are three ways of estimating the frequency response of a plant and they are:

- *Experimental* [8]: The open loop plant is subjected to a sinusoidal test signal of a given frequency and the magnitude and phase of the output sinusoidal signal are recorded. This experiment is repeated for different frequencies in the input sinusoidal signal. Such experiments can give very accurate frequency behavior of the plant, but this method can be tedious, time consuming, expensive and may not be feasible under most circumstances (particularly for chemical plants). An extension of this basic method is pulse testing.
- *Signal processing methods*: Thses methods were described in Chapter 2.
- *Mathematical models*: Mathematical models described in the ensuing sections can generate frequency response data. But the accuracy of this method depends on the quality of the estimated mathematical model. This technique is chiefly used to validate the quality of a mathematical model and/or to design robust/stable controllers.

Frequency response models are generically denoted by $G(z)$ as mentioned in Chapter 2.

3.4 Step response and FIR models

A step response model is obtained by providing a step input to the plant whereas a FIR model is obtained by subjecting a plant to an impulse input or by differencing the step model coefficients. Mathematically these models are represented as:
Step-response model:

$$G(q) = \sum_{i=1}^{N} s_i \Delta u(t-i) \tag{3.2}$$

Impulse-response model:

$$G(q) = \sum_{i=1}^{N} f_i u(t-i) \tag{3.3}$$

where s_i and f_i are the i^{th} step and impulse response coefficient respectively and from equations-(3.2) and (3.3) it is obvious that such models are ***unstructured***, ***non-parametric***,

input/output and ***output error*** type (OE models are explained later in Section 3.5). These models are also ***black-box*** type because no ***a priori*** process information is required to obtain these models.

Several industrial predictive controllers used in the process industries are based on FIR/step-response models. Generally a large number of coefficients is required to describe the dynamics of a process. FIR or step-response models are typically of the order of 20 to 60 or even more, depending on the sampling time and nature of the process dynamics. However, these models are particularly good in capturing unusual dynamics of the process.

3.4.1 Estimation of Step/FIR models

Step/FIR models can be estimated in the following ways:

- *Experimental*: Step response coefficients can be estimated by subjecting the plant to a step input. Similarly FIR coefficients can be obtained by subjecting the plant to an impulse input. But such tests are limited by:
  - In presence of noise and disturbances, the step/FIR coefficients are adversely affected and they do not represent the true (noise free) plant dynamics.
  - For high SNR in a plant, step tests are feasible, but impulse tests may still be infeasible because: (a) FIR coefficients are very small ($f_i = \Delta s_i$) and they usually get masked by the noise; and (b) it may be difficult and impractical to inject an impulse input to a plant (especially for chemical plants).

  Therefore FIR coefficients are usually derived using: $f_i = \Delta s_i$. Step/FIR coefficients can also be obtained using least squares methods, but such a method is limited by: (a) input step lengths must be long enough to cover the entire plant dynamics; (b) there can be numerical problems because the number of parameters is large; and (c) these parameters are very sensitive to plant noise and disturbances.

- *From other models*: Other forms of mathematical models can be converted to step or FIR models. Such conversions are usually done to compare quality of the estimated models.

3.5 Transfer function models

All discrete linear noise-free processes with a ***zero-order hold*** (ZOH) can be expressed in the following ***transfer function*** form:

$$G(z) = \frac{B(z)}{A(z)} \tag{3.4}$$

where $z = e^{j\omega T_s}$ for the frequency $\omega = [0, \omega_N]$ and where $\omega_N = \pi/T_s$ is the *Nyquist frequency* and T_s is the sampling time. In the time domain it is also called an ARX[5] model and is given by

$$y(t) = G(q)u(t) = \frac{B(q)}{A(q)}u(t) \tag{3.5}$$

$G(q)$ in (3.5) is the ***transfer operator*** [9] and $B(q) = b_1q^{-1} + b_2q^{-2} + \cdots + b_{nb}q^{-nb}$ and $A(q) = 1 + a_1q^{-1} + a_2q^{-1} + \cdots + a_{na}q^{-na}$. It should be noted that if a ZOH is included in the model, a unit delay is implicit in $G(q)$ as shown in the above expression for $B(q)$. A more generalized transfer function model

$$\begin{aligned} A(q)y(t) &= q^{-d}\frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}\xi(t) \\ &= q^{-d}\frac{B(q)}{F(q)}u(t) + v(t) \end{aligned} \tag{3.6}$$

accounts for the disturbances where $v(t) = C(q)\xi(t)/D(q)$ in the above expression is referred as the colored noise, process delay is denoted by d and ZOH is implicit in $B(q)$. The orders $\delta(\cdot)$ of different polynomials in equation-(3.6) are: $na = \delta(A(q))$, $nb = \delta(B(q))$, $nc = \delta(C(q))$, $nd = \delta(D(q))$ and $nf = \delta(F(q))$. All other forms of transfer function models such as ARX, ARMAX, ARIMAX, BJ etc and even FIR model are subclasses of the model represented by equation (3.6)[6] [2]. From equation (3.6) it is obvious that transfer function models are ***structured***, ***parametric***, ***I/O type*** and ***black box*** models. Furthermore, the form given by equation (3.6) is defined as the ***equation error*** (EE) model if $/delta(A(q)) \geq 1$, otherwise for $A(q) = 1$ it is called a Box-Jenkin's (BJ) model which is also defined as an ***output error*** (OE) model.

3.5.1 Estimation of transfer function models

An ARX model is obtained from equation 3.6 by setting $F(q) = C(q) = D(q) = 1$ and the parameters $\theta = [B(q), A(q) - 1]$ for such a model can be estimated from a batch of I/O data $z(N) = [u(t), y(t), t = 1 \ldots N]$ by minimizing [1]

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}[y(t) - \phi^T(t)\theta(t-1)]^2 \tag{3.7}$$

where $\phi^T(t) = [u(t-1), \ldots, u(t-nb), \ \ -y(t-1), \ldots, -y(t-na)]$. The solution of equation-(3.7) is given by the Gauss-Markov least-squares (LS)[7] solution as:

---

[5] ARX stands for Auto regressive model with eXogenous input.

[6] Where: AR=Auto Regressive, MA=Moving Average, X=eXogenous, BJ=Box Jenkins, I=Integrated

[7] LS method can be traced back to Gauss in 1809. Markov's contribution came much later [1, 10].

$$\hat{\theta} = \left[\sum_{t=1}^{N} \frac{1}{N}\phi(t)\phi(t)^T\right]^{-1}\left[\sum_{t=1}^{N}\phi(t)y(t)\right] \tag{3.8}$$

The above method is called the batch LS (BLS) method because a batch of data are used to obtain parameter estimates. There are also recursive methods of parameter estimates used in adaptive-control but this thesis is restricted to the use of batch identification techniques. It is also possible to simultaneously estimate the parameters (in equation (3.8)) and loss functions (in equation (3.7)) for different orders of ARX models using the Augmented UD Identification (AUDI) method [11]. The use of the AUDI method is illustrated by applying it to industrial case studies in Sections 3.10 and 3.11. For colored noise *i.e.* $v(t) = C(q)/D(q)\xi(t) = H(q)\xi(t)$ the LS solution can be obtained by filtering $z(N)$ by $H^{-1}(q)$. Since $H(q)$ is unknown, $\hat{H}(q)$ is used to approximate $H(q)$. The filter $\hat{H}^{-1}(q)$ can either be obtained by trial and error or if it can be somehow approximated. The choice of the filter becomes relatively simple if either $C(q) = 1$ or $D(q) = 1$.

A more general prediction error method (PEM) can be applied to $z(N)$ to estimate the parameters of equation-(3.6) [1, 2, 5]. PEM employs standard optimization methods to minimize

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\varepsilon^2(t,\theta) \tag{3.9}$$

to obtain its solution in an efficient manner. In equation-(3.9), $\varepsilon(t,\theta) = y(t) - \phi^T(t)\theta(t-1)$ is the *prediction error*. Newtonian algorithms such as:

$$\theta(t+1) = \theta(t) - [V_N''(\theta(t))]^{-1}[V_N'(\theta(t))]^T \tag{3.10}$$

can be used to update θ where

$$\begin{aligned}
V_N'(\theta(t)) &= \left.\frac{dV_N(\theta)}{d\theta}\right|_{\theta=\theta(t)} = -\frac{1}{N}\sum_{t=1}^{N}\lambda(t,\theta)\varepsilon(t,\theta) \\
V_N''(\theta(t)) &= \left.\frac{d^2V_N(\theta)}{d\theta^2}\right|_{\theta=\theta(t)} \\
&= \frac{1}{N}\sum_{t=1}^{N}\lambda(t,\theta)\lambda^T(t) - \frac{1}{N}\sum_{t=1}^{N}\frac{d\lambda(t,\theta)}{d\theta}\varepsilon(t,\theta)|_{\theta=\theta(t)} \qquad (3.11)
\end{aligned}$$

and where $\lambda^T(t,\theta) = -d\varepsilon(t,\theta)/d\theta = d(\hat{y}(t|\theta))/d\theta$ at $\theta = \theta(t)$ and $\hat{y}(t|\theta) = \phi^T(t)\theta(t-1)$. Close to the solution θ_0, $\frac{1}{N}\sum_{t=1}^{N}\frac{d\lambda(t,\theta)}{d\theta}\varepsilon(t,\theta) \approx 0$ in equation-(3.11) as $\varepsilon(t,\theta) \approx e_0(t)$ is constant. Therefore the Hessian of $V_N(t)$ can simply be approximated by:

$$V_N''(\theta(t)) = \frac{1}{N}\sum_{t=1}^{N}\lambda(t,\theta)\lambda^T(t)|_{\theta=\theta(t)} \tag{3.12}$$

which avoids the computation of $d\lambda(t,\theta)/d\theta$. A combination of equations-(3.10) to (3.12) is also known as the *Gauss-Newton* method which gives a positive definite Hessian thus guaranteeing convergence to a minima. The approximation-(3.12) does not affect the search procedure significantly because the effect of Hessian is not important when: (a) the solution is far from θ_0; and (b) $V_N(\theta(t) \to \theta_0)$ cannot be well approximated by a quadratic function [2]. If equation-(3.12) approaches singularity, then a *regularization* technique such as *Levenberg-Marquardt* method can be used *i.e.* $\delta\mathbf{I}$ is added to the RHS of equation-(3.12) where δ is small positive scalar and $\mathbf{I}$ is the identity matrix [2]. Gradients $\lambda(t,\theta)$ in equation-(3.11) for equation-(3.6) are shown to be [1, 2, 5]:

$$
\begin{aligned}
\frac{\partial\hat{y}(t|\theta)}{\partial a_k} &= -\frac{D(q)}{C(q)}y(t-k) \\
\frac{\partial\hat{y}(t|\theta)}{\partial b_k} &= \frac{D(q)}{C(q)F(q)}u(t-k) \\
\frac{\partial\hat{y}(t|\theta)}{\partial c_k} &= \frac{1}{C(q)}\varepsilon(t-k,\theta) \\
\frac{\partial\hat{y}(t|\theta)}{\partial d_k} &= -\frac{1}{C(q)}v(t-k,\theta) \\
\frac{\partial\hat{y}(t|\theta)}{\partial f_k} &= -\frac{D(q)}{C(q)F(q)}w(t-k,\theta)
\end{aligned}
\tag{3.13}
$$

where $w(t,\theta) = B(q)u(t)/F(q)$, $v(t,\theta) = A(q)y(t) - w(t,\theta)$, $\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta) = D(q)v(t,\theta)/C(q)$ and $y(t) = [(C(q) - A(q)D(q))/C(q)]y(t) + D(q)B(q)u(t)/C(q)F(q) + \xi(t)$. Therefore a regressor corresponding to $\theta^T = [A(q)-1,\ B(q),\ F(q)-1,\ C(q)-1,\ D(q)-1]$ for equation-(3.6) becomes $\phi^T(t) = [-y(t-1),\ldots,-y(t-n_a),\ u(t-1),\ldots,u(t-n_b),\ -w(t-1,\theta),\ldots,-w(t-n_f,\theta),\ \varepsilon(t-1,\theta),\ldots,\varepsilon(t-n_c,\theta),\ -v(t-1,\theta),\ldots,-v(t-n_d,\theta)]^T$.

In order to avoid local minimas in the Gauss-Newton approach, an effort is made to obtain good initial values[8]. It is however shown that there are no local minimas for: (a) ARMA models; (b) BJ models with $n_f = 1$, and (c) OE models ($A(q) = C(q) = D(q) = 1$) if $u \in \mathcal{N}(\mu,\sigma^2)$; and for ARMAX models local minimas are rare [2, 1]. Model validation techniques outlined in Section 3.9 are used to check if the solution has reached a local minima. The PEM method is also used (in fact essential) in closed loop identification, as will be seen in Chapter 5.

3.6 Orthonormal function models

A linear stable transfer function $G(z)$ can be expressed as a weighted sum of the orthonormal functions [4, 12]:

---

[8] Good initial values can be generated by first approximating an ARX model followed by approximating a noise model.

$$G(z) = \sum_{k=1}^{\infty} g_k \Psi_k(z) \tag{3.14}$$

where the coefficients g_k, $(k = 1, 2, 3, ...)$ are called the weights or gains of the orthonormal functions $\Psi_k(z)$. The scope of the above equation-(3.14) is defined using a number of mathematical premises that are based on functional analysis [13, 14, 15]. The following paragraph highlights some of these mathematical presumptions for the orthonormal functions; however they are pursued with a greater depth in Chapter 4 (orthonormal function models are the main topic of discussion in Chapter 4).

The function $\Psi_k(z)$ in equation (3.14) is defined as the orthonormal basis function and this function is said to be orthonormal if $< \Psi_i(z), \Psi_j(z) >= \delta_{ij}$ where $< \cdot >$ is the inner product and δ_{ij} is the Kronecker delta function (*i.e.* $\delta_{ij} = 1$ if $i = j$ otherwise $\delta_{ij} = 0$ for $i \neq j$). The basis function in $\Psi_k(z)$ in equation (3.14) is chosen such that it reflects the unit delay that is implicit in the discrete transfer function $G(z)$. If the basis signal $\psi_k(t)$ for the orthonormal block $\Psi_k(z)$ be given by: $\psi_k(t) = \Psi_k(q)u(t)$, then the true plant output can be expressed as:

$$y(t) = \sum_{k=1}^{\infty} g_k \psi_k(t) = \sum_{k=1}^{\infty} g_k \Psi_k(q) u(t) \tag{3.15}$$

In practice only a finite number of orthonormal functions are required to approximately represent the process dynamics. Therefore the model output for an N^{th} order orthonormal series is given by:

$$\hat{y}(t) = \sum_{k=1}^{N} g_k \psi_k(t) \tag{3.16}$$

There are several types of orthonormal functions such as Laguerre, Kautz, Markov (or FIR), Markov-Laguerre functions etc. used to describe the process dynamics. These functions are discussed in greater details in Chapter 4. Only the Markov-Laguerre model is mentioned in this chapter because this model is used for system identification case studies in the subsequent sections.

A Markov-Laguerre model is obtained by combining the Markov and Laguerre functions. Markov functions are used to capture the fast, delayed or inverse response behavior of the plant just like an FIR model whereas the Laguerre functions are used to capture the low frequency dynamics of the plant. For overdamped systems an approximate knowledge of the location of the real pole a, $(\mid a \mid < 1)$ in a plant is required to obtain Laguerre function model, therefore orthonormal models such as the Laguerre function are called a *grey-box* models. The real pole a in the Laguerre series represents the dominant time constant of the process. The discrete Laguerre functions for a real pole in the z-domain are defined as [4, 16]:

$$\Psi_k(z) = L_k(z) = \frac{\sqrt{1-a^2}}{z-a}\left(\frac{1-az}{z-a}\right)^{k-1} \tag{3.17}$$

The above Laguerre filter pertains to discrete models where a unit delay due to the ZOH is implicit in the model. Similarly the basis function for a Markov function is $\Psi_k(z) = z^{-k}$ or $\Psi_k(q) = q^{-k}$. Markov and Laguerre functions are combined as

$$\hat{G}(z) = \sum_{i=1}^{d} h_i q^{-i} u(t) + \sum_{i=1}^{N} g_i \Psi_i(q) u(t-d) \tag{3.18}$$

to obtain Markov-Laguerre models where d is the number of Markov coefficients, N is the number of Laguerre functions, h_i is the i^{th} Markov coefficient, g_i is the i^{th} Laguerre coefficient and $\Psi_i(q)$ corresponds to the i^{th} Laguerre function as shown in equation-(3.17).

3.6.1 Estimation of Markov-Laguerre models

A batch of I/O data can be used to estimate the Markov-Laguerre parameters by using the BLS method shown by equation-(3.8). For a specified number of Markov (d) and Laguerre (N) coefficients, the data vector

$$\phi(t)^T = [u(t-1), \ldots, u(t-d), l_1(t-d), \ldots, l_N(t-d)]^T \tag{3.19}$$

is used to estimate the parameters $\theta = [h_1, \ldots, h_d, g_1, \ldots, g_N]$ (using BLS) where $l_k(t-d) = L_k(q)u(t-d)$ and $L_k(q)$ is the same as $\Psi_k(q)$ as shown in equation (3.17).

3.7 State space models

The knowledge of different dynamic modes of a system that are internal to the process are used to develop a state space (SS) model. Usually a state space model gives a more detailed picture of the process dynamics than the I/O models. But it is often tedious and difficult to develop SS models especially for chemical plants, hence the emphasis of this thesis is on utilizing I/O models for identification and control.

The design of controllers is usually simplified when SS models are used instead of some other form of model such as the orthonormal models. Therefore models such as orthonormal models are converted to SS form prior to the controller design. Infact all the models described in Sections 3.4 to 3.6 can be converted into the SS form. It should be noted that the SS models obtained using model-transformation most often do not correspond to physical states of the process. Therefore such translated SS models are also described as pseudo-SS models and they are used because of mathematical convenience.

The conversion of orthonormal models to SS form is described in Chapter 6. Further discussion on SS models is limited because they are not used in this chapter.

3.8 Confidence bounds

Confidence bounds for the estimated parameters obtained using BLS are based on the assumption: $\hat{\theta} - \theta_0 \in \mathcal{N}(0, P_N)$ where P_N is the covariance matrix given by:

$$P_N = \sigma^2[\Phi_N^T\Phi_N]^{-1} = \sigma^2\left[\sum_{t=1}^{N}\phi(t)\phi^T(t)\right]^{-1} \tag{3.20}$$

where θ_0 is the true parameter vector and $\sigma^2 = V_N(\theta_0)$ is true variance of the noise which is unknown. The relation between σ^2 and $\hat{\sigma}^2$ is given by [2]:

$$\hat{\sigma}^2 = \frac{N}{N-\varpi}V_N(\hat{\theta}) \tag{3.21}$$

where the degree of freedom (dof) ϖ is the dimension of $\hat{\theta}$. $\hat{\sigma}^2$ is given by the following χ^2 distribution [2]:

$$(N-\varpi)\frac{\hat{\sigma}^2}{\sigma^2} \in \chi^2_{N-\varpi}(\alpha) \tag{3.22}$$

where α is the confidence level discussed earlier in Chapter 2. The normality assumption in $\Delta\theta = \hat{\theta} - \theta_0$ translates into the following χ^2 distribution [2]:

$$\varpi * \Delta\theta^T P_N^{-1}\Delta\theta \in \chi^2_{\varpi}(\alpha) \tag{3.23}$$

Since σ^2 (which is unknown) is implicit in P_N in equation (3.23), a better estimate of the confidence bound is obtained by substituting equation (3.21) in equation (3.23) and dividing it by $\hat{\sigma}^2$ (from equation (3.22)) to produce [2]

$$\frac{\Delta\theta^T[\Phi_N^T\Phi_N]\Delta\theta}{\hat{\sigma}^2} \in F_{\varpi,N-\varpi}(\alpha) \tag{3.24}$$

where $F_{\varpi,N-\varpi}(\alpha)$ is as discussed in Chapter 2. Therefore the confidence limit for the i^{th} parameter becomes:

$$\Delta\theta_i \leq \sqrt{\hat{\sigma}^2[\Phi_N^T\Phi_N]_{ii}^{-1}F_{\varpi,N-\varpi}(\alpha)} \tag{3.25}$$

where $[\Phi_N^T\Phi_N]_{ii}^{-1}$ is the i^{th} diagonal element of $[\Phi_N^T\Phi_N]^{-1}$. Equation-(3.24) describes a constant probability density contour which is surface of a hyperellipsoid centered at θ_0 in parameter space [10] whose *length of semi axes* l_i are:

$$l_i = \sqrt{\frac{\hat{\sigma}^2F_{\varpi,N-\varpi}(\alpha)}{\lambda_i}} \tag{3.26}$$

and their *directions* are given by e_i where λ_i and e_i are the i^{th} eigenvalue and eigenvector of $[\Phi_N^T\Phi_N]$ [2, 10]. In general the axes of confidence hyperellipsoids are given by $\pm l_i\sqrt{e_i}$ and its volume ϑ is shown to be [10]:

$$\vartheta = \kappa_{\varpi}\sqrt{\hat{\sigma}^2[\Phi_N^T\Phi_N]^{-1}F_{\varpi,N-\varpi}(\alpha)} \tag{3.27}$$

where $\kappa_{\varpi} = 2\pi^{\varpi/2}/\varpi\Gamma(\varpi/2)$ and $\Gamma(\cdot)$ is the Gamma function. If $\hat{\theta}$ corresponds to the FIR parameters, then confidence limit for the estimated model spectrum becomes: [17]:

$$\frac{\Delta\Theta^T[H^T[\Phi_N^T\Phi_N]H]^{-1}\Delta\Theta}{\hat{\sigma}^2} \leq F_{\varpi,N-\varpi}(\alpha) \tag{3.28}$$

where

$$H = \begin{bmatrix} H_{real} \\ H_{imag} \end{bmatrix} = \begin{bmatrix} cos(\omega T) & cos(2\omega T) & \dots & cos(\varpi\omega T) \\ -sin(\omega T) & -sin(2\omega T) & \dots & -sin(\varpi\omega T) \end{bmatrix} \tag{3.29}$$

is the basis matrix that converts FIR parameters to frequency domain and $\Theta = H^T\theta$. From equation-(3.28), axes for the confidence ellipsoid at each frequency can then be shown to be:

$$\Delta\Theta_i = \sqrt{\hat{\sigma}^2 F_{\varpi,N-\varpi}(\alpha)\lambda_{ii}(H^T[\Phi_N^T\Phi_N]^{-1}H)} \tag{3.30}$$

3.9 Steps in model identification and validation

Usually the following steps are involved in the identification and validation of a SISO linear model from open loop I/O data of the plant [1, 2, 18, 19]:

1. *Experiment design*: The first logical step in system ID is to collect a set of process I/O data by providing excitation to the plant. This excitation signal can be either stochastic or deterministic. As mentioned in Chapter 2, the stochastic signals such as the white noise or RBS are characterized by a certain probability distribution whereas the deterministic signals depend a lot on the design parameters.Irrespective of the nature of inputs, they should: (a) have sufficient magnitude such that the SNR of the output signal is rich enough and yet it does not violate the process operating conditions; and (b) persist for sufficient duration to meet the identification requirements.

   The advantage of deterministic input is that its spectrum can be shaped according to the ID needs. Some of the parameters that are useful for designing an input are: *a priori* knowledge of the plant dominant time constant, sampling time, settling time, cross over frequency, desired closed loop specifications such as the closed loop bandwidth, disturbance dynamics, etc. The use of these variables to design an input are illustrated by the following examples:

   - The Schroeder-phased signals [20] are dependend on the choice of a number of the above stated variables. Such signals are a combination of the sinusoids and therefore: (a) they can be used to obtain a spectrum of the input as desired; and (b) such a spectrum can be determined without using classical signal processing methods[9].

---

[9] This type of input has not been used in this thesis.

- For control applications, a good estimate of the model around the cross over frequency is important. Therefore a good estimate of the model around the cross over frequency can be ensured by designing an input signal that is rich around these frequencies.
- Sometimes it is desired to obtain a good estimate of the model around the lower frequencies as well. For such cases 'stretched' RBS signals can be used, where the stretch factor is dependent on the dominant time constant of the process.

If possible, steady state data should also be collected to assess the structure of noise in the plant. Usually two sets of experiments are performed, where one set is used for ID and the other set is used for model validation.

2. *Data processing and preliminary data analysis*: The experimental or given data set should then be zero centered and any outliers should be eliminated. If the knowledge of the approximate structure of the noise is available then it can be used to linearize the data or else the data are linearized and if possible made stationary by suitable prefiltering. Results of Chapter 2 are then used for the following preliminary analysis of the processed data:

    - *Delay*: Cross-correlation and visual methods are used to assess process delay and the direction of gain. Large data samples *i.e.* 1024 or more points are required for good assessment using cross-correlation techniques.
    - *Coherency*: Squared coherency tests are performed to check the level of noise and the extent of correlation between process input and output.
    - *Process spectrum*: Signal processing methods with appropriate smoothing are used to assess the Bode characteristics of the process.

3. *Choice of model and its identification method*: The processed data from the preliminary analysis stage can be further filtered depending on the choice of model type and structure. If the plant has more measurement noise or if it is suspected that the noise is directly added to the output rather than to the process, then OE, BJ or orthonormal function models are expected to identify the plant better. In the case of process noise or if the noise is suspected to have an EE form, then ARX, ARMAX, ARIMAX etc. models are expected to identify the plant better. For general model structures such as BJ, ARIMAX etc models, a PEM method is suggested whereas for orthonormal models or ARX models BLS identification method is adequate. The AUDI method can be applied to ARX/ARMAX models to assess process delays, order and determine parameters for different orders of the model. The Akaike Information Criteria (AIC) can also be used to confirm the choice of an appropriate model order.

4. *Model validation*: The estimated model is validated in the time and frequency domain and also by using residual analysis as shown below:
    - *Time domain validation*: The quality of the estimated model is visually assessed by superimposing model output on the plant output, where the set of plant data used for time domain validation is usually different from the data set used for identification. Since the model output does not consider noise and other disturbances, it may not seem to fit well with the plant output when the data are corrupted by noise and disturbances. However disturbances and noise can be accounted in the model by fitting one step ahead model prediction on the plant data. One step ahead prediction is also known as model output with feedback.
    - *Frequency domain validation*: The estimated model is visually evaluated by superimposing its Bode plots on the Bode plots estimated using signal processing methods from the plant I/O data. The two most obvious points to check are the steady state gain and phase shift due to an error in the estimated delay.
    - *Residual analysis*: The quality of the model can be assessed by examining the whiteness of the residual *i.e.* $\epsilon(t) = y(t) - \hat{y}(t)$. If the residual $\epsilon(t)$ is white, then the estimated model is said to be of good quality. The residual needs appropriate filtering in the presence of colored noise. The whiteness of $\epsilon(t)$ can be tested using the ACF/PACF plots.
5. *Re-identification*: If model validation is unsatisfactory, then the whole identification process is repeated (*i.e.* Go to step-2) by suitably changing the model delay, order, structure, type etc as the case may be.

The above steps are illustrated via example case studies in the ensuing two sections.

3.10 Case-I: Shell benchmark problem

This section discusses the application of ID methods outlined in Sections 3.3 to 3.9 to estimate dynamic models for the Shell benchmark problem introduced in Section 3.1. In essence this section: (a) summarizes a revised version of the solution that was proposed by the U of A group[10] [7] and in addition (b) discusses the use of BJ and orthonormal function models to represent the plant dynamics.

The layout of this section is as follows: (a) subsection 3.10.1 outlines the Shell benchmark problem; (b) subsection 3.10.2 gives the overall identification strategy; (c) subsection 3.10.3 describes identification of noise models; (d) preliminary data analysis is described in subsection 3.10.4; (e) different models used for ID are discussed in subsection 3.10.5; and (f) this is followed by residual analysis and model evaluation in subsections 3.10.6 and 3.10.7 respectively.

---

[10] The U of A group was represented by a team of graduate students and staff from the 'Process Control Group', Department of Chemical Engineering [7].

3.10.1 Problem description

The Shell plant is in the form of FORTRAN executable code, that very closely emulates a real 2 × 2 industrial distillation column. The computer code[11] was provided instead of real industrial data because: (a) it provides flexibility for designing the plant input; and (b) with simulated data different identification methods can be compared as the underlying model is known to the head evaluator; which is not the case with real data.

The Shell plant data are highly corrupted by noise and unmeasured disturbances. The base level of noise which poses a tough challenge for model identification is designated as the 100% noise case. The option is also provided to select 20% and 50% of the base level noise to reduce the difficulty of the identification exercise.

Note 3.2 *Based on the "rules" established for this competition, each level of noise was treated as an independent problem. Results for one level of noise were not used to influence the outcome of another level of noise.*

Note 3.3 *The definition of SNR is modified in this case study i.e. each level of noise* 20%, 50% *and* 100% *is designated as a different SNR for the sake of simplicity.*

Operating conditions

The industrial column has 35 real trays with feed entering at tray 5 (numbered from the top tray) and it is assumed that the feed flow is constant. This column is intended to prevent a light key component in the feed from contaminating the bottom product (bottoms impurity is X).

A steam heated stab-in reboiler provides the necessary heat Q to the column and it is also used to control X. The reboiler heat duty is controlled by manipulating the steam flow rate and the bottoms level is controlled separately by manipulating the reboiler flow rate.

An overhead partial condenser uses plant cooling water to condense the overhead products and manipulate the overhead pressure P. However, the column throughput presently exceeds the original design, hence heat removal is a constraint and to partially account for this the cooling water flows at maximum capacity at all times. An accumulator separates uncondensed overheads D and the condensed overhead is sent back as a reflux. The reflux rate is manipulated separately by an accumulator level controller. Process objectives, operating conditions and constraints forthe key manipulated and controlled variables are summarized in the following table [6]:

---

[11] The computer code was in an object form. Therefore the original plant dynamics were not known to the participants.

The manipulated variables are Q and D which are used to achieve the above mentioned objectives. Furthermore, the specified sampling time for the plant is $T_s = 5$ (min) and the number of data points must not exceed a limit of 4096 which corresponds to approximately 14 days of operation. The main objectives for this problem are:

- Identify the process models and discuss the effect of unmeasured disturbances on model identification.
- Design a PID controller which would tie X with Q and P with D.
- Design a multivariable, linear, predictive controller which would decouple X and P.

This chapter discusses only the first objective *i.e.* model identification. The other two objectives (*i.e.* controller design) are reported in [7].

3.10.2 Overall solution procedure

Based on Section 3.9, the following provides a brief layout of the steps involved in the identification of the Shell benchmark problem:

1. *Identification of noise model*: The noise model is estimated from the steady state plant data.
2. *Plant excitation and preliminary data analysis*: The plant is excited with D±10 and Q±200 to generate sets of I/O data. A preliminary analysis is performed on the I/O data to ascertain delays, coherency and cross correlation in the data.
3. *Model identification*: Knowledge of the noise model is used to pre-filter the data in order to improve the identification of the plant dynamic models.
4. *Model validation*: Estimated models are validated in the time/frequency domain and also by performing residual tests.
5. *Re-identification*: Go to step 3 and suitably change the model order, delay, filter or any other relevant parameters to confirm that they are optimal. Usually experiments are not repeated during the model re-identification stage.

The noise pattern actuated by the plant program code is controlled by a 'seed' that generates a particular sequence of random numbers. This 'seed' in turn is dependent on the clock time of the computer that triggers different noise sequences each time the program is run and thus allowing it to emulate an industrial scenario. Therefore for each set of identification experiment the noise sequence in the plant data are different. However to compare the effect of SNR on the identified models, the same random 'seed' is used for different SNRs for a particular identification experiment.

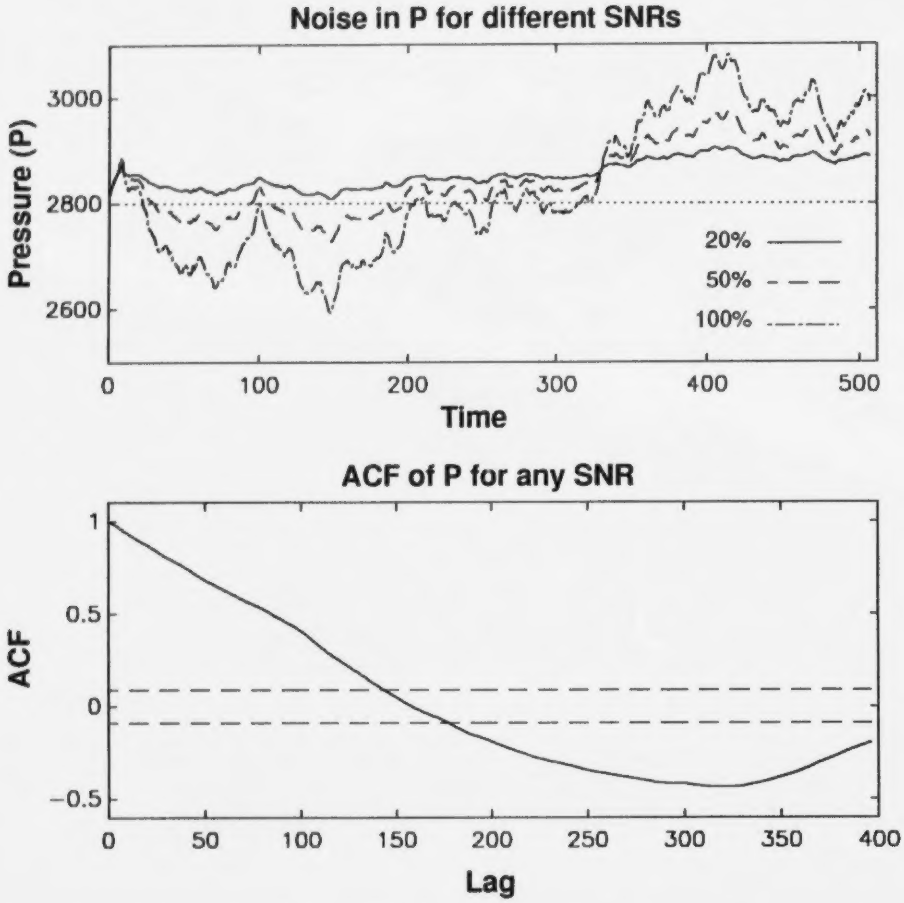


Figure 3.1: Noise in P and its ACFs for different SNRs. Top: Time series data. Bottom: ACF.

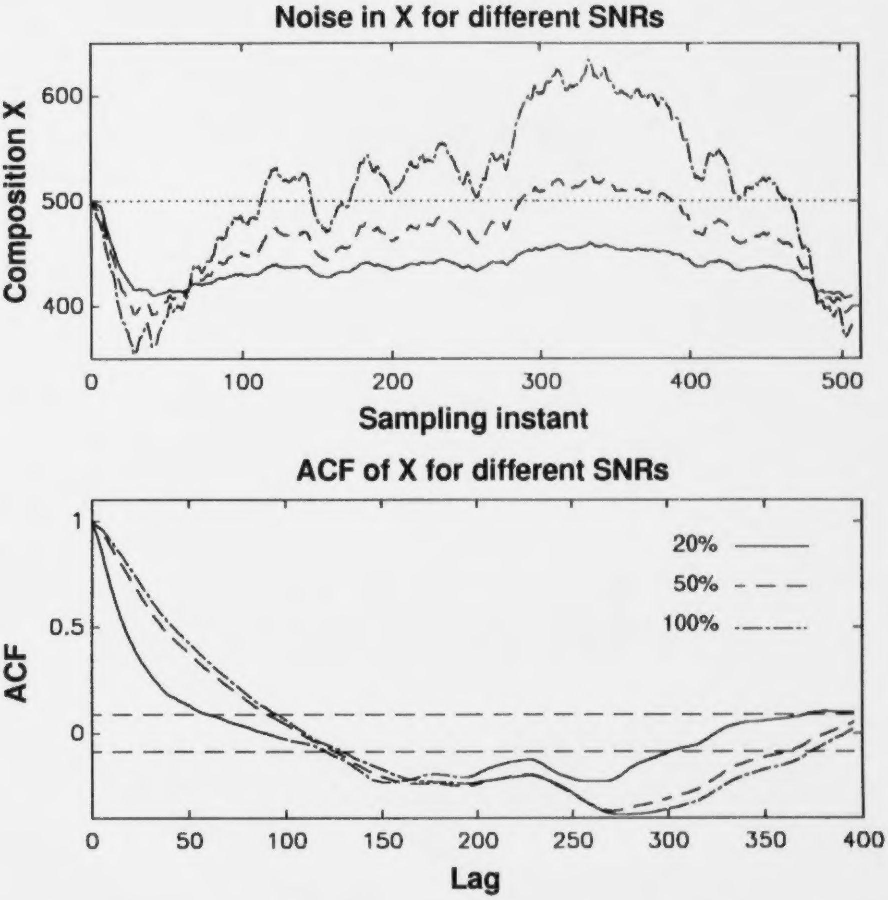


Figure 3.2: Noise in X and its ACFs for different SNRs. Top: Time series data. Bottom: ACF.

3.10.3 Identification of noise models

In order to estimate the noise structure, a record of P and X are obtained with the manipulated variables D and Q constant at their nominal operating points. It is possible to perform this experiment because there are no restrictions on the input design. The results of this open loop steady state experiment are shown in Figures 3.1 and 3.2 for all levels of noise. These figures also show autocorrelation functions (ACFs) of P and X sequences for different SNRs. The ACF plots in Figures 3.1 and 3.2 reveal that:

- There is significant auto-correlation in P and X, thus indicating that the noise *may* have an AR structure.
- The auto- correlation in P is the same irrespective of the SNR.
- The auto-correlation in X is different for different SNRs. However in view of Note-3.2, this information was not exploited to ascertain the nature of noise in X.

The AR noise in P and X are confirmed by their partial autocorrelation functions (PACFs) which indeed reveals an AR(2) structure (*i.e.* there are two poles, but this figure is not included). Furthermore, the strong correlations in the ACF of P and X at higher lags indicate that one of the poles might be located near the unit circle. If the dominant pole for the AR(2) noise is assumed to be at 1 (*i.e.* integrated noise $\equiv 1/\Delta$), then first differencing (*i.e.* $\Delta = 1 - q^{-1}$) of the data should linearize this data and make it stationary.

The need for first differencing of the data are confirmed by the ACF and PACF plots of ΔP and ΔX in Figure 3.3 which shows that the noise structure is indeed integrated AR(1). If the noise in P and X is denoted by v_1 and v_2 respectively then from Figure 3.3 it can be determined that:

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \frac{\xi(t)}{\Delta(1-d_1q^{-1})} \\ \frac{\xi(t)}{\Delta(1-d_2q^{-1})} \end{pmatrix} = \begin{pmatrix} \frac{\xi(t)}{\Delta D_1} \\ \frac{\xi(t)}{\Delta D_2} \end{pmatrix} \tag{3.31}$$

where $d_1 = 0.5745$ and values of d_2 are 0.7535, 0.6296, and 0.5896 for SNRs 20%, 50% and 100% respectively. D_1 and D_2 in equation-(3.31) are the noise polynomials and $\xi(t)$ is the white noise.

3.10.4 Plant dynamic data and preliminary analysis

Since the plant is 2×2, (*i.e.* two outputs (P and X) and two inputs (D and Q)) and the noise is defined by equation-(3.31), the plant model has the following OE form:

$$\begin{pmatrix} P(q) \\ X(q) \end{pmatrix} = \begin{pmatrix} G_{11}(q) & G_{12}(q) \\ G_{21}(q) & G_{22}(q) \end{pmatrix} \begin{pmatrix} D(q) \\ X(q) \end{pmatrix} + \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix} \tag{3.32}$$

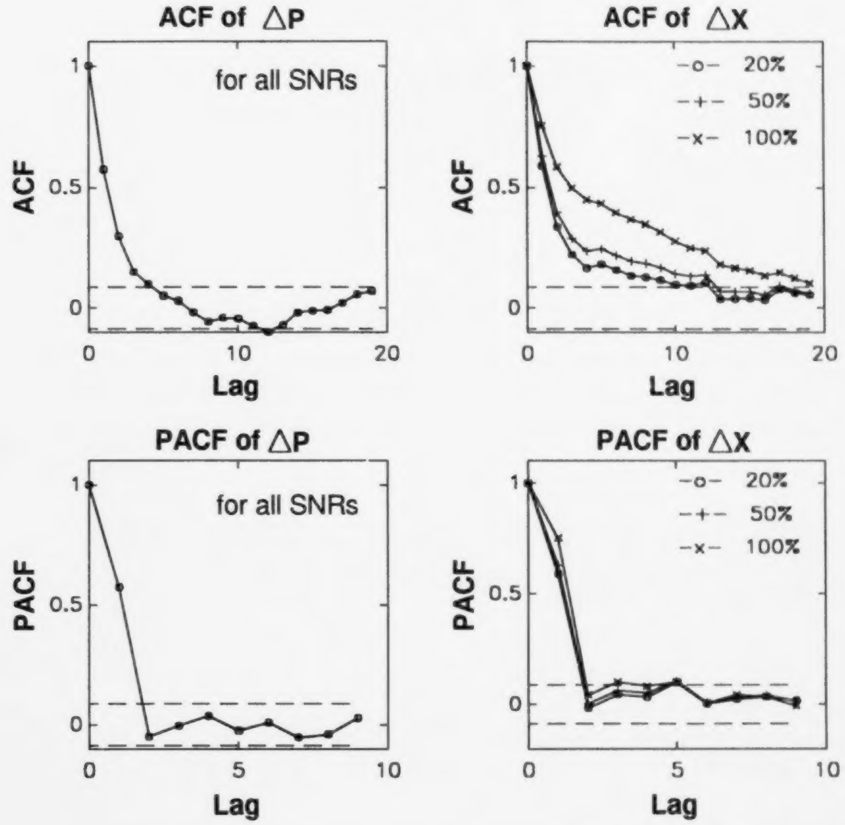


Figure 3.3: ACFs and PACFs of differenced noise data for different SNRs. Top-Left and Right: ACF of ΔP and ΔX. Bottom Left and Right: PACF of ΔP and ΔX.

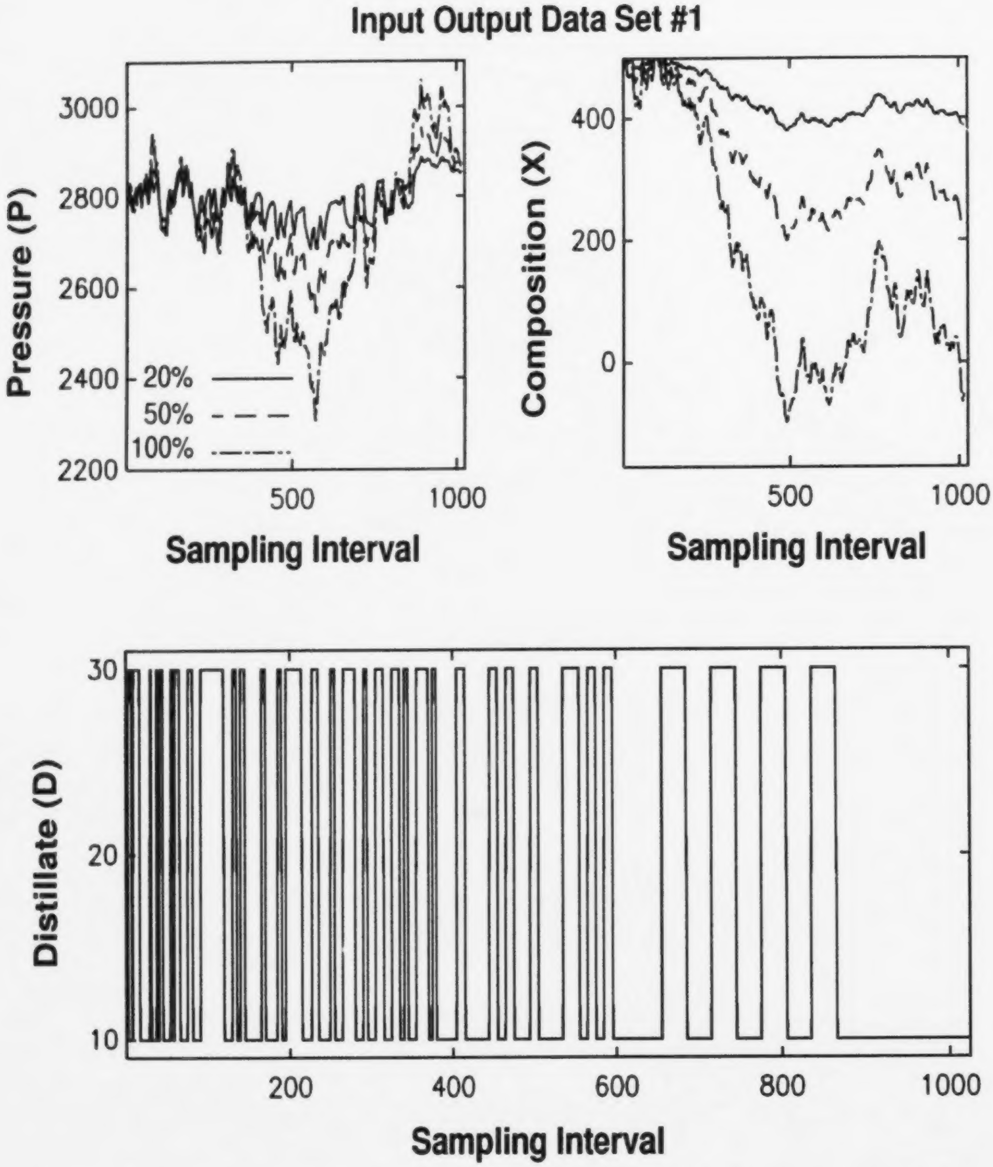


Figure 3.4: Plant responses for different SNRs when subjected to excitation in D.

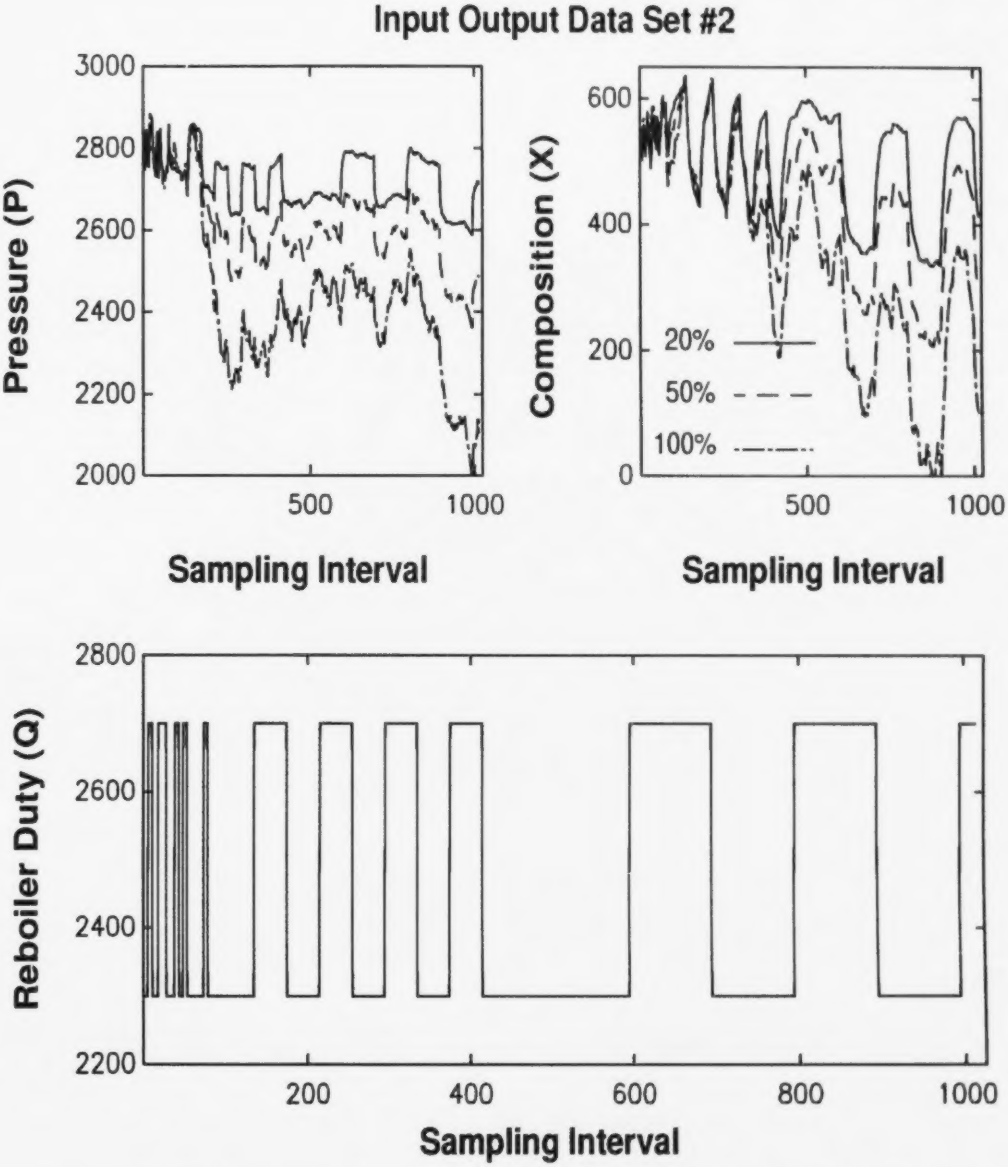


Figure 3.5: Plant responses for different SNRs when subjected to excitation in Q.

where for an ARX structure $G_{ij}(q) = B_{ij}(q)/F_{ij}(q)$ and for orthonormal models it is $G_{ij}(q) = \sum_{k=1}^{N_{ij}} g_{ijk}\Psi_{ijk}(q)$.

The models in equation-(3.32)[12] are estimated by subjecting the plant to stretched RBS excitation in such a way that: (a) they do not violate the plant operating conditions/constraints; and (b) the excitation is rich enough to excite all frequency modes of the plant. The advantage of flexibility in the input design and the sampling duration was used to provide two SISO excitation signals to the plant, each lasting 1024 sampling periods (*i.e.* $3\frac{1}{2}$ days). Figure 3.4 shows the plant response with different SNRs when subjected to the RBS excitation in D holding Q constant. Similarly Figure 3.5 shows the plant response with different SNRs by providing RBS excitation in Q and keeping D constant. (Note: It is interesting to note that X goes below zero for 100% noise case as shown in Figure 3.4. In practice such a situation will of course never be encountered.)

A preliminary estimate of the coherency between the I/O data[13] is obtained from the squared coherency spectrum as shown for different channels in Figure 3.6. The plots of squared coherency spectrum in Figure 3.6 show that:

- The overall squared coherency for 20% noise is high for all the channels except for X/D, thus indicating good model identifiability for all these channels except for X/D.
- The squared coherency for 50% noise is very close to the 20% case for P/D and P/Q, hence for these channels the estimated models are expected to be close for both 20% and 50% noise cases.
- The squared coherency is significantly lower for the 100% noise case and therefore difficulties would be expected in the model identification.

The cross correlation function (CCF) analysis of the process data in Figure 3.7 for different levels of noise shows that:

- There are no delays (and not even a ZOH!) for channels P/D and P/Q.
- The channel X/Q has a delay of 7 sampling intervals *i.e.* 6 delays and 1 due to ZOH.
- There is no cross correlation in the channel X/D.

From the preliminary analysis based on the data in Figures 3.6 and 3.7 it can be said, with a 95% certainty, that there are no dynamics in the channel X/D *i.e.* $G_{21}(q) = 0$, therefore no further attempt is made to identify dynamics for this channel.

---

[12] It was later revealed that the channel X/Q was nonlinear, however around an operating point it behaved linearly. Hence the assumption of a linear model for this channel worked well.

[13] Note that the original data are detrended or mean centered before identification.

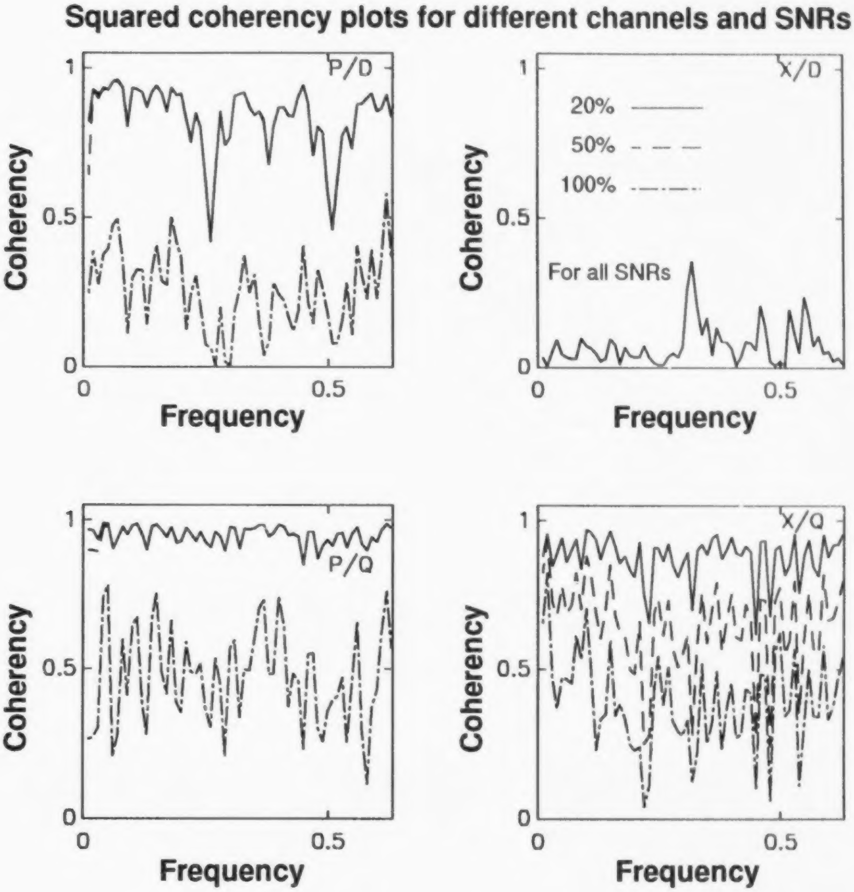


Figure 3.6: Squared coherency functions for different channels and different SNRs. Top-Left: Channel P/D. Top-Right: Channel X/D. Bottom-Left: Channel P/Q. Bottom-Right: Channel X/Q.

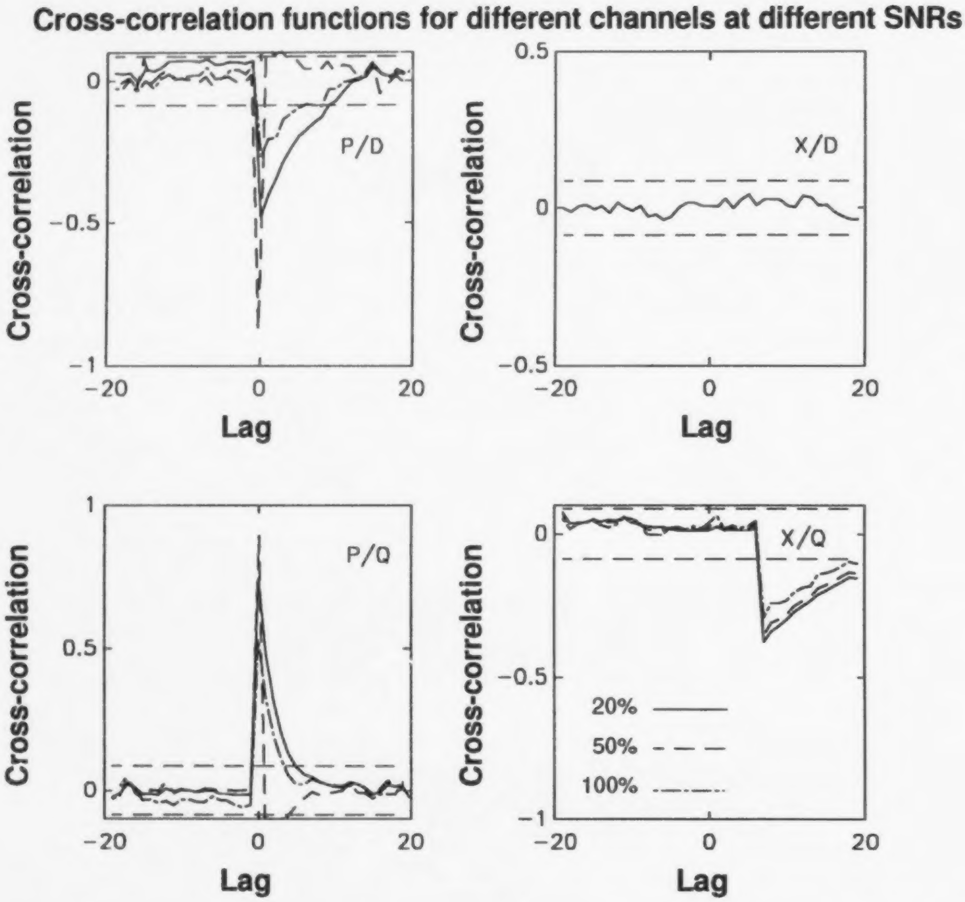


Figure 3.7: Cross correlation functions for different channels and different SNRs. Top-Left: Channel P/D. Top-Right: Channel X/D. Bottom-Left: Channel P/Q. Bottom-Right: Channel X/Q.

3.10.5 Model estimation

For this case study, the following models and estimation procedures are considered:

- ARX $\Longleftrightarrow$ BLS/AUDI
- BJ $\Longleftrightarrow$ PEM
- Markov-Laguerre $\Longleftrightarrow$ BLS

Based on the model structure given by equation-(3.32), the transfer function for each channel can be represented by the following BJ model:

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{1}{\Delta D(q)}\xi(t) \tag{3.33}$$

ARX Model

The ARX model is estimated by first transforming the OE structure in equation-(3.33) into the EE form

$$F(q)y(t) = B(q)u(t) + \frac{F(q)}{\Delta D(q)}\xi(t) \tag{3.34}$$

and filtering it by $\Delta/\hat{D}(q)$ as

$$F(q)\frac{\Delta y(t)}{\hat{D}(q)} = B(q)\frac{\Delta u(t)}{\hat{D}(q)} + \frac{F(q)}{D(q)\hat{D}(q)}\xi(t) \tag{3.35}$$

where $\hat{D}(q)$ is chosen such that $F(q) \approx D(q)\hat{D}(q)$ to make the noise term white. The data vector consisting of the filtered I/O data *i.e.* $u^f(t) = \Delta u(t)/\hat{D}(q)$ and $y^f(t) = \Delta y(t)/\hat{D}(q)$ is then used to estimate the ARX parameters using the BLS/AUDI method. The use of AUDI method in Banerjee *et. al.* [7] reveals that a first order model is adequate to describe the process dynamics. The estimated ARX models for different SNRs summarized in Tables 3.2 to 3.4 indicate the following for each channel:

- Channel P/D: With a decrease in SNR: (a) the estimated ARX parameters become more biased and (b) the steady state gain is increasingly overestimated *i.e.* the true gain is −4.49 whereas for 20%, 50% and 100% noise levels the respective estimated gains are −4.85, −5.8 and −7.38.
- Channel P/Q: Different levels of noise influence the estimated parameters and gains in the same way as for P/D. The true gain for this case is 0.31 whereas for 20%, 50% and 100% noise levels the estimated gains are 0.309, 0.357 and 0.46 respectively.

- Channel X/Q: With the decrease in SNRs there is no significant change in the estimated parameters however the estimated gains are overestimated *i.e.* the true gain is -0.499 whereas for 20%, 50% and 100% noise levels, the estimated gains are -0.53, -0.55 and -0.61 respectively.

It may be noted that no ZOH was considered in the models corresponding to the P/D and P/Q channels. The lack of ZOH in these channels was independently confirmed from their phase spectrum plots obtained using signal processing methods (shown later in Figures 3.8 and 3.9), and the cross-correlation tests (in Figure 3.7).

BJ Model

The PEM method was used to estimate the parameters of the BJ model by setting $A(q) = C(q) = 1$ in equation-(3.6) and filtering equation-(3.33) by Δ so that:

$$\Delta y(t) = \frac{B(q)}{F(q)}\Delta u(t) + \frac{\xi(t)}{D(q)} \tag{3.36}$$

BJ models estimated for different SNRs are summarized in Tables 3.2 to 3.4 and for each channel they indicate the following trend was exhibited:

- Channel P/D: The estimated parameters are fairly consistent with the change in noise levels, however the gains are slightly underestimated in this case *i.e.* the estimated gains for 20%, 50% and 100% noise levels are -4.59, -4.49 and -4.36 respectively (true gain = -4.49).

- Channel P/Q: The estimated parameters are marginally affected by the change in noise levels and here also the gains are underestimated for 20%, 50% and 100% noise levels the estimated gains are 0.299, 0.282 and 0.255 respectively (true gain = 0.31).

- Channel X/Q: The estimated parameters are fairly consistent and the SNRs marginally affect them. However the gains are overestimated *i.e.* for 20%, 50% and 100% noise levels, the estimated gains are -0.52, -0.54 and -0.59 respectively (true gain = -0.499).

Markov-Laguerre Model

The OE structure in equation-(3.33) is retained to estimate the Markov-Laguerre model. In order to minimize the effect of noise in the parameter estimation, equation-(3.33) is filtered by $\Delta\hat{D}(q)$ as:

$$\Delta\hat{D}(q)y(t) = \frac{B(q)}{F(q)}(\Delta\hat{D}(q)u(t)) + \frac{\hat{D}(q)}{D(q)}\xi(t) \tag{3.37}$$

No Markov parameters are considered for channels P/D and P/Q because there are no delays in these channels, whereas for the channel X/Q, 6 Markov parameters are considered to accomadate the process delay. The ZOH is not inherent in the Laguerre functions for channels P/D and P/Q because these channels react instantaneously to process excitation as analyzed in Section 3.10.4.

The estimated parameters g_i of Markov-Laguerre models shown in Tables 3.2 to 3.4 indicate that they are quite immune to the changes in the SNRs. The Laguerre poles in all these cases are approximately equal to the average of poles estimated using the ARX and BJ models. The steady state gains of the Markov-Laguerre models are compared by plotting step responses as shown later in subsection 3.10.8.

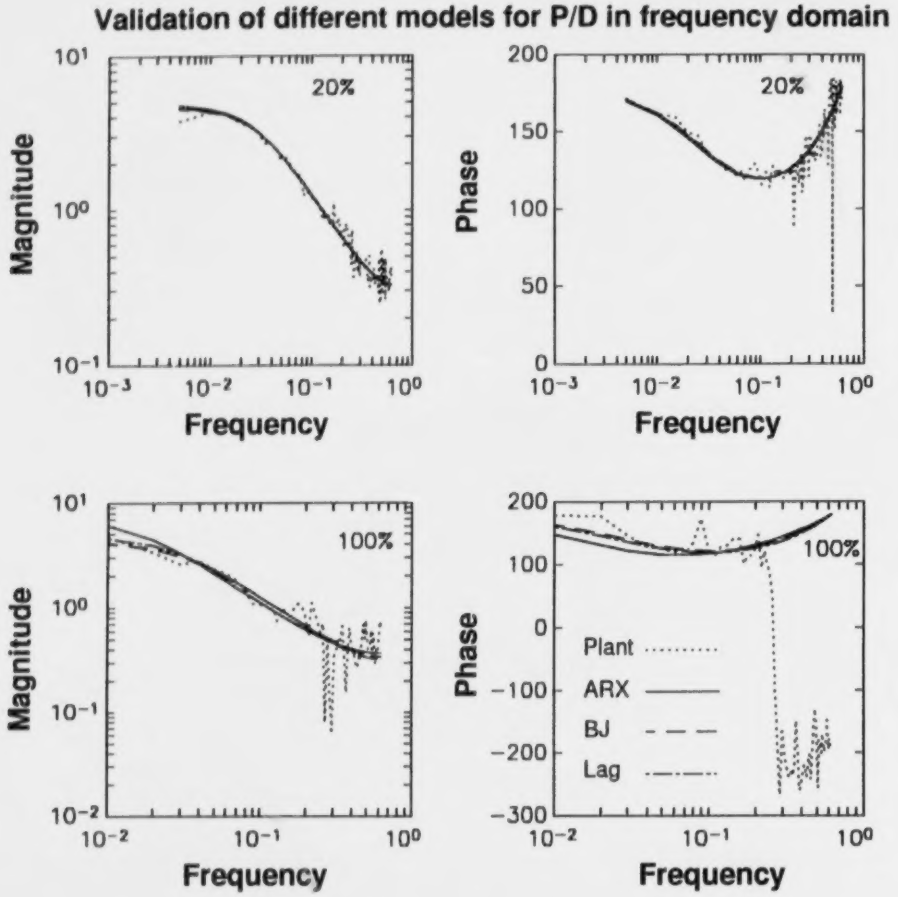


Figure 3.8: Frequency domain model validation for channel P/D. Top-Left: Magnitude spectrum for 20% noise. Top-Right: Phase spectrum for 20% noise. Bottom-Left: Magnitude spectrum for 100% noise. Bottom-Right: Phase spectrum for 100% noise.

3.10.6 Model validation

The estimated models shown in Tables 3.2 to 3.4 are visually validated in the frequency and time domains in this subsection. For the sake of brevity all figures related to the validation studies show only the 20% and 100% noise cases. The results for the 50% noise case fall between the 20% and 100% noise levels and they are not reported in this chapter for the sake of brevity.

Frequency domain model validation

Frequency domain models are validated by visually comparing the Bode plots of the models with the Bode plots obtained using the signal processing methods as discussed in Chapter 2. The Bode plots obtained using the signal processing methods are also referred to as the *spectral plots* for brevity.

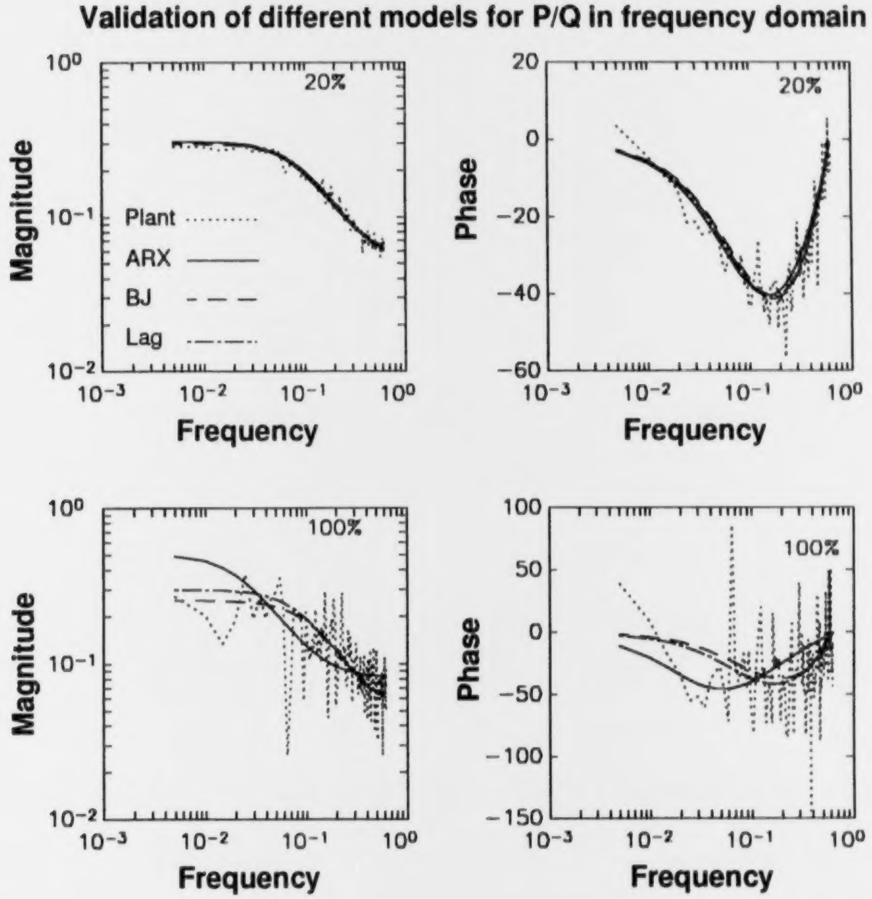


Figure 3.9: Frequency domain model validation for first order models for channel P/Q. Top-Left: Magnitude spectrum for 20% noise. Top-Right: Phase spectrum for 20% noise. Bottom-Left: Magnitude spectrum for 100% noise. Bottom-Right: Phase spectrum for 100% noise.

Figures 3.8 to 3.10 shows the validation of estimated models using the Bode plots[14]. For the case of 20% noise, both magnitude and phase plots of the models match very well with the spectral plots for all the channels. For the case of 100% noise the following are observed:

- The spectral estimates show significant variance because of the lower coherency even though more data windows are used for this case.
- Unusual behavior is observed in the phase spectral plot for the channel P/D at the higher frequencies. This unusual behavior is attributed to the high level of noise in the data.
- All models seem to fit well in the frequency domain for the channel X/Q. For the channel P/D and P/Q, the Bode plots for BJ and Markov-Laguerre models almost seem to be in agreement with each other and to a large extent they also seem to be following the trends in the spectral plots. However the Bode plots for the ARX models for the channels P/D and P/Q show significant deviation from the spectral plots as well as from the Bode plots corresponding to the BJ and Markov-Laguerre models.

Time domain model validation

In time domain the models are visually validated by comparing the estimated output of the model with the plant output. The estimated output can be of two types *i.e.* (a) model output and (b) one step ahead predicted output. Model output $\hat{y}(t)$ is given by:

$$\hat{y}(t) = \hat{G}(q)u(t) \tag{3.38}$$

where $\hat{G}(q)$ is the estimated plant model and the one step ahead predicted output $\hat{y}(t+1|t)$ is given by:

$$\hat{y}(t+1|t) = \phi^T(t)\hat{\theta} \tag{3.39}$$

where $\phi(t)$ is the data vector and $\hat{\theta}$ is the estimated parameters mentioned in Section 3.5. For transfer function models it is straightforward to calculate $\hat{y}(t+1|t)$, however for the orthonormal function models (*i.e.* Markov-Laguerre model in this case) a transformation to the ARX form is required to estimate $\hat{y}(t+1|t)$. The expression for the ARX model obtained from the Markov-Laguerre model is given in Section 6.2 in Chapter 6; has been used here. Equation-(3.39) can also be used to quantify the quality of the model in terms of the residual *i.e.* $V_N(\hat{\theta}) = \frac{1}{N}\sum_{i=1}^{N}[y(i) - \hat{y}(i+1|i)]^2$.

---

[14] In Figures 3.8 and 3.9, the phase spectrums show unusual behavior because of the lack of ZOH in the corresponding plant transfer functions.

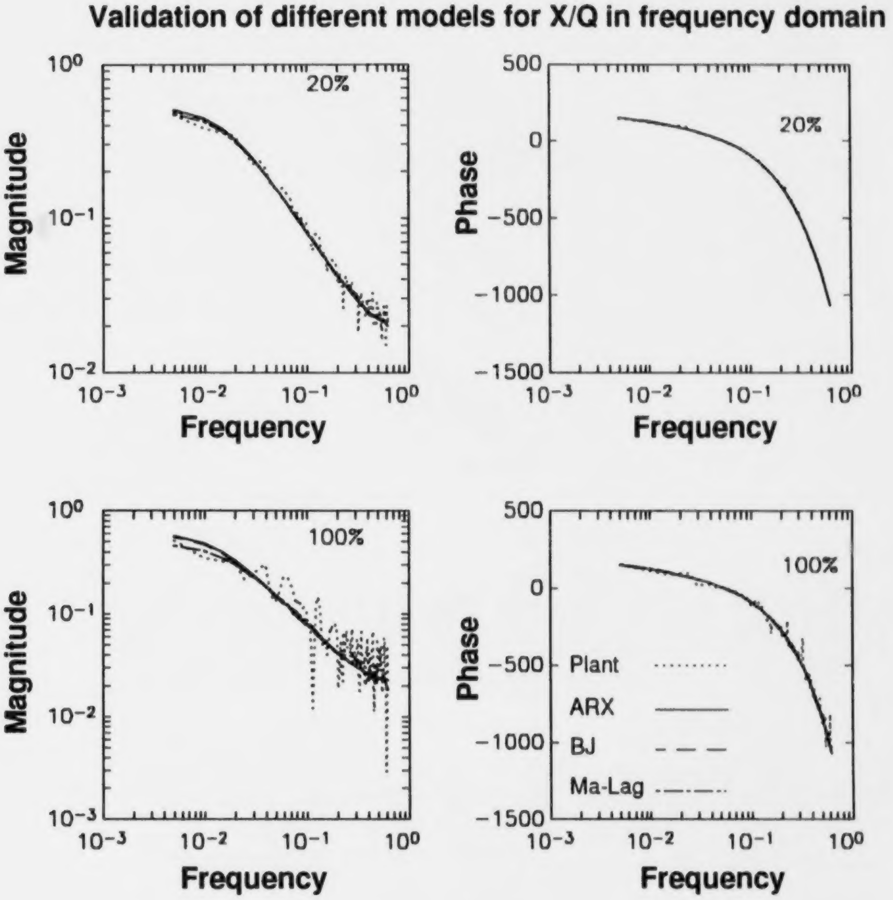


Figure 3.10: Frequency domain model validation for channel X/Q. Top-Left: Magnitude spectrum for 20% noise. Top-Right: Phase spectrum for 20% noise. Bottom-Left: Magnitude spectrum for 100% noise. Bottom-Right: Phase spectrum for 100% noise.

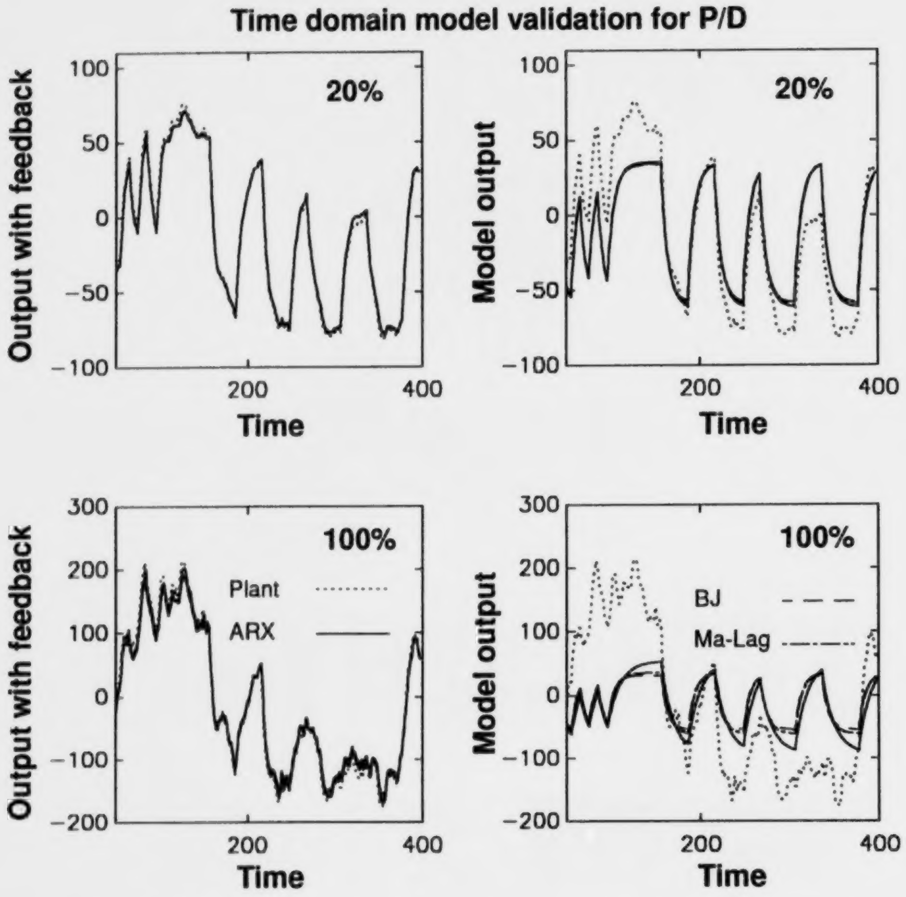


Figure 3.11: Time domain model validation for channel P/D. Top-Left: Model output with feedback for 20% noise. Top-Right: Model output without feedback for 20% noise. Bottom-Left: Model output with feedback for 100% noise. Bottom-Right: Model output without feedback for 100% noise.

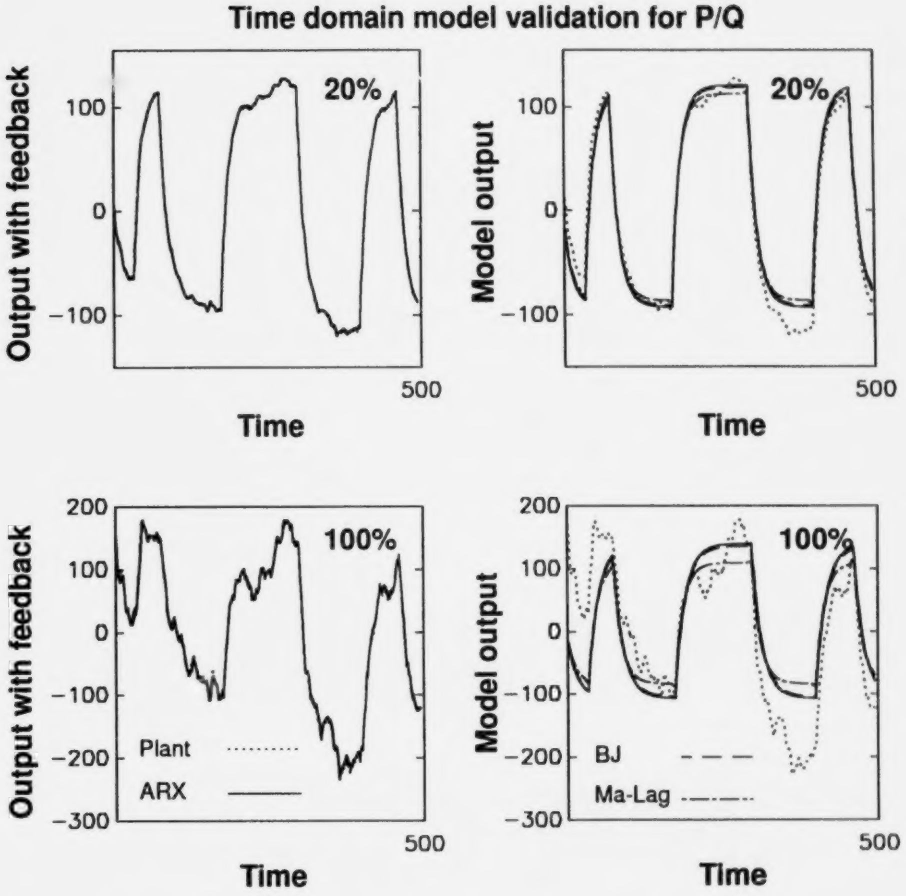


Figure 3.12: Time domain model validation for channel P/Q. Top-Left: Model output with feedback for 20% noise. Top-Right: Model output without feedback for 20% noise. Bottom-Left: Model output with feedback for 100% noise. Bottom-Right: Model output without feedback for 100% noise.

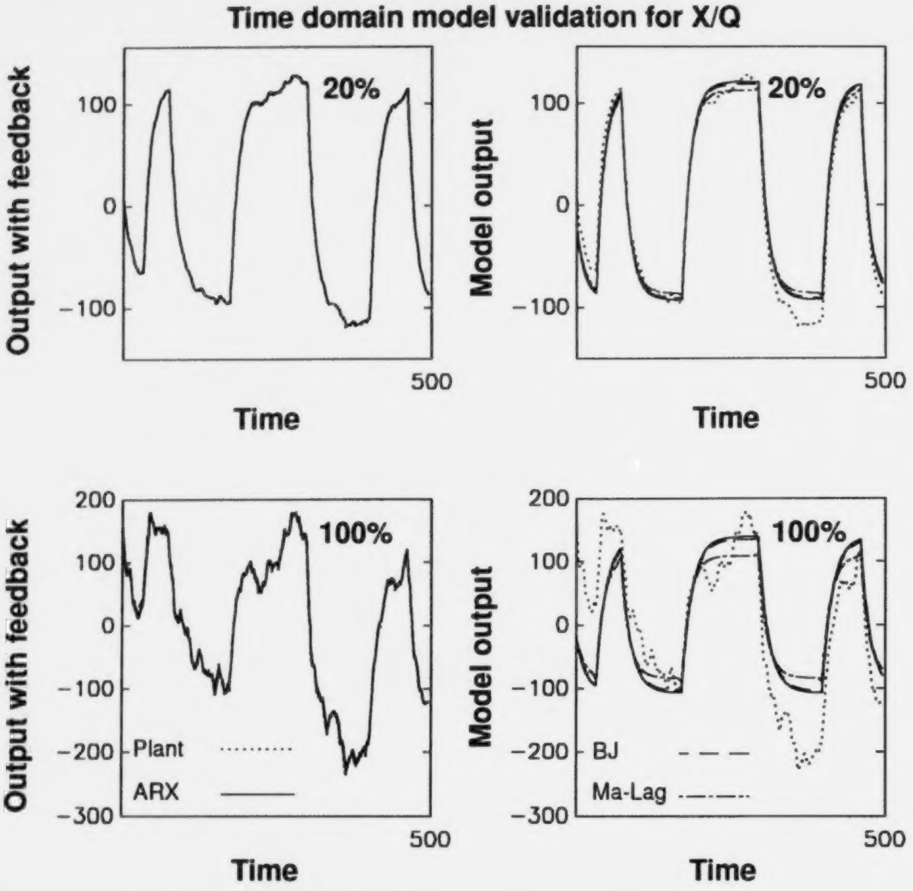


Figure 3.13: Time domain model validation for channel X/Q. Top-Left: Model output with feedback for 20% noise. Top-Right: Model output without feedback for 20% noise. Bottom-Left: Model output with feedback for 100% noise. Bottom-Right: Model output without feedback for 100% noise.

If the process output is corrupted by noise or disturbances then $\hat{y}(t)$ may not appear to match well with the output $y(t)$ because equation-(3.38) does not utilize any noise/disturbance information that are buried in $y(t)$. On the other hand $\hat{y}(t+1|t)$ usually matches well with $y(t)$ because it utilizes past process outputs and therefore $\hat{y}(t+1|t)$ is also defined as ***output with feedback***.

Time domain model validations using $\hat{y}(t)$ and $\hat{y}(t+1|t)$ for channels P/D, P/Q and X/Q are illustrated in Figures 3.11 to 3.13 for 20% and 100% noise levels. The model output $\hat{y}(t)$ is compared with the zero-centered and detrended (linear trend is removed) process output $y(t)$ whereas $\hat{y}(t+1|t)$ is compared with only zero-centered $y(t)$.

For 20% noise, the Figures 3.11 to 3.13 show that: (a) the match between $\hat{y}(t+1|t)$ and $y(t)$ is better than between $\hat{y}(t)$ and $y(t)$ for all channels; and (b) for X/Q, $\hat{y}(t)$ is in

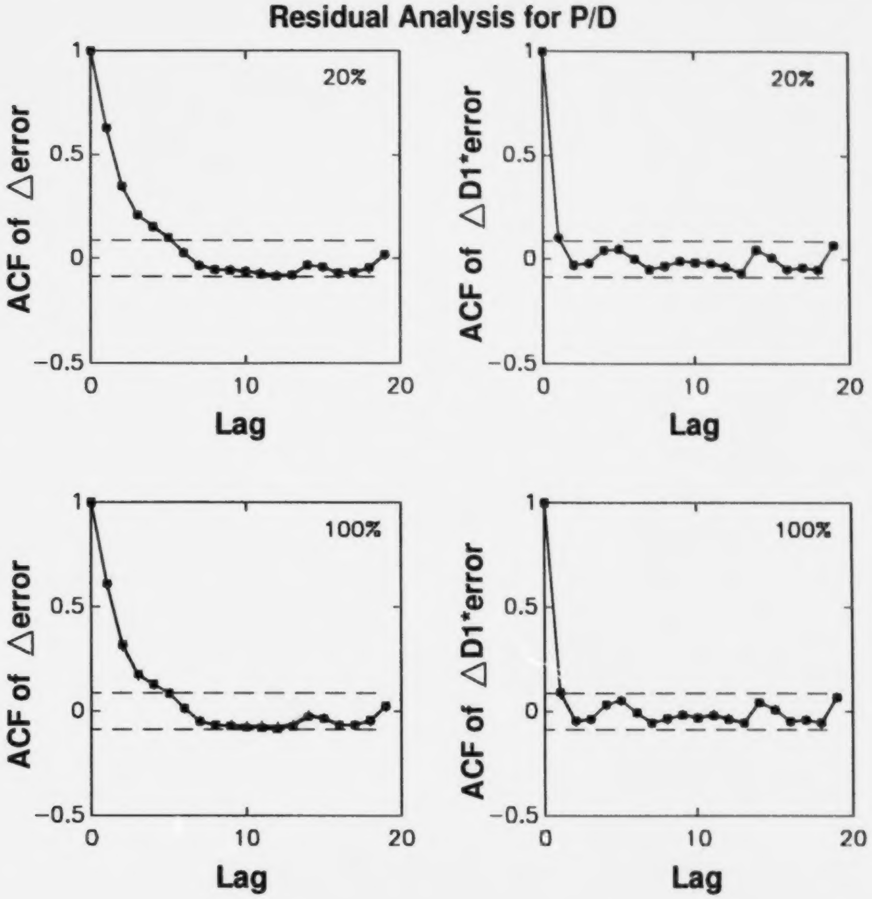


Figure 3.14: Residual analysis for channel P/D. Top-Left: ACF(Δe) for 20% noise. Top-Right: ACF($\hat{D}_1(q)\Delta e$) for 20% noise. Bottom-Left: ACF(Δe) for 100% noise. Bottom-Right: ACF($\hat{D}_2(q)\Delta e$) for 100% noise.

good agreement with $y(t)$, thus indicating that this channel is signal-rich.

For case of 100% noise, Figures 3.11 to 3.13 indicate that: (a) $\hat{y}(t)$ is significantly different from $y(t)$, because of the significant level of disturbaces; and (b) $\hat{y}(t+1|t)$ appears to match $y(t)$.

3.10.7 Residual analysis

Figures 3.14 to 3.16 show the residual analysis for all the three channels for 20% and 100% noise levels where the residual $e(t)$ is given by $e(t) = y(t) - \hat{y}(t)$. Based on the approximate knowledge of the noise model given by equation-(3.31), it is expected that the residuals would be white if $e(t)$ is filtered according to:

$$e^f(t) = \Delta\hat{D}(q)e(t) \tag{3.40}$$

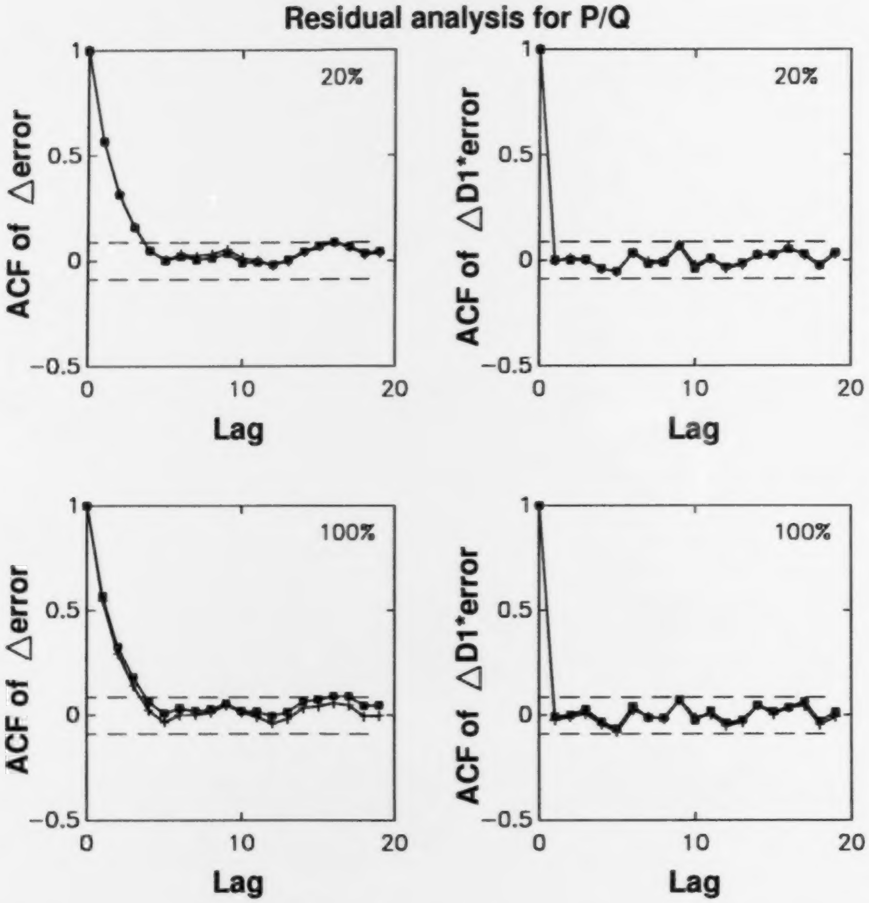


Figure 3.15: Residual analysis for channel P/Q. Top-Left: ACF(Δe) for 20% noise. Top-Right: ACF($\hat{D}_1(q)\Delta e$) for 20% noise. Bottom-Left: ACF(Δe) for 100% noise. Bottom-Right: ACF($\hat{D}_2(q)\Delta e$) for 100% noise.

The ACF plots of residuals in Figures 3.14 to 3.16 show that:

- $\Delta e(t)$ is not white for all the channels, models and levels of noise thus indicating that $\xi(t)/\Delta$ is only a partial representation of the noise structure.
- The residuals $e^f(t)$ obtained according to equation-(3.40) are white for channels P/D and P/Q thus confirming that the structure of noise model selected is correct for the output P.
- The residual $e^f(t)$ is not perfectly a white noise signal for channel X/Q, which means that the noise structure selected for X is imperfect. However from subsection 3.10.6 it is noted that $\hat{G}(q)$ fits well in the time and frequency domain for X/Q, hence this channel does not require re-identification. Nevertheless, for the sake of accuracy an alternate model structure (such as EE) could be selected for the re-identification of X/Q channel.

3.10.8 Step responses

The step responses of the different models and the spectrum of the estimated model-plant mismatch (MPM) are depicted in Figures 3.17 to 3.19 for all the three channels, models and for the two levels of noise. For different channels these plots show the following:

- P/D: Laguerre model gives the best fit for 20% and 100% noise, which is followed by the BJ and ARX models. The mismatch in the steady state gain for the ARX model is very significant for 100% noise. The level of noise is so significant that the spectral plots of the MPM are significantly overestimated. For the sake of clarity, the spectral plot of MPM is shown only for the ARX model.
- P/Q: Around steady state, the Laguerre and ARX models give a better fit for 20% noise, whereas for 100% noise, the Laguerre model gives the best estimate. The BJ model is marginally underestimated whereas the ARX model is significantly overestimated. However at the higher frequencies BJ model gives the best performance as can be seen from the MPM plots. For the sake of clarity, the spectral plot of MPM is shown only for the ARX model.
- X/Q: The performance of all the models is comparable for both cases of noise, however around steady state the performance of Laguerre model is better. The spectral plot of MPM corresponds to the ARX model.

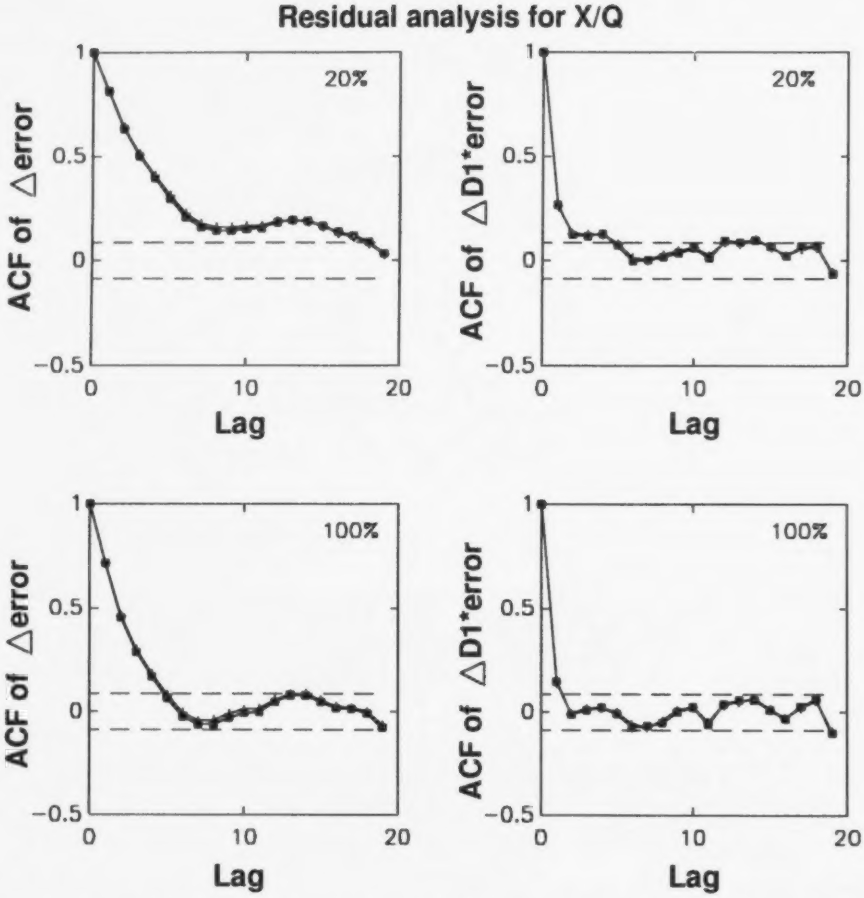


Figure 3.16: Residual analysis for channel X/Q. Top-Left: ACF(Δe) for 20% noise. Top-Right: ACF($\hat{D}_1(q)\Delta e$) for 20% noise. Bottom-Left: ACF(Δe) for 100% noise. Bottom-Right: ACF($\hat{D}_2(q)\Delta e$) for 100% noise.

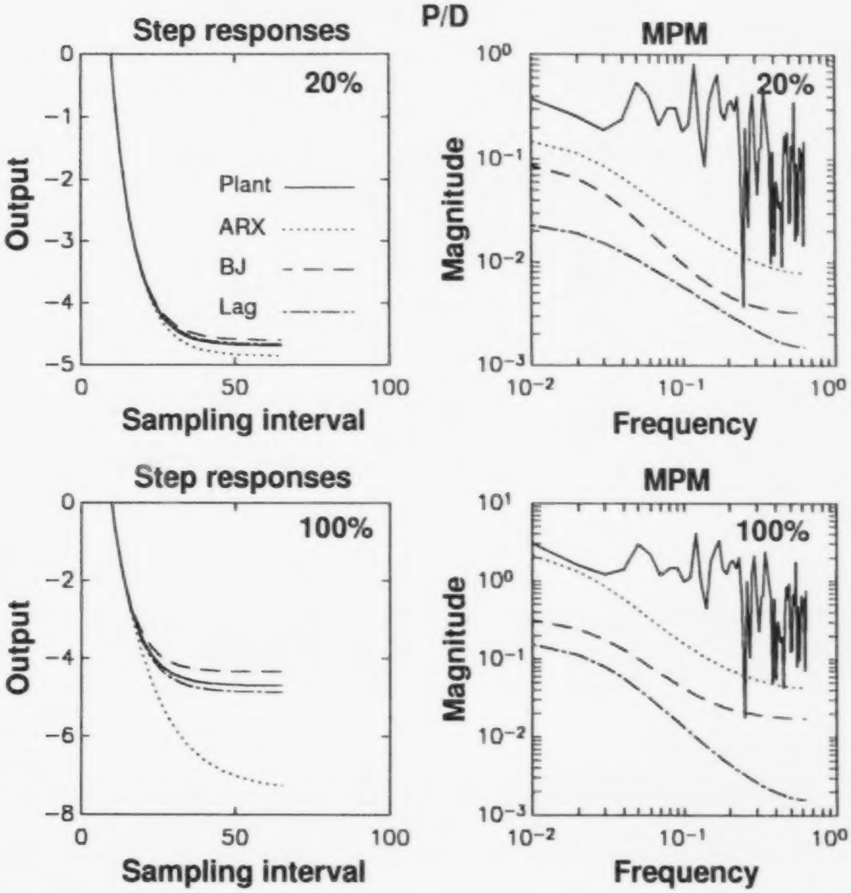


Figure 3.17: Step responses and MPM for different identified models for channel P/D. Top-Left: Step responses of models identified for 20% noise. Top-Right: Step responses of models identified for 100% noise. Bottom-Left: MPM for models identified for 20% noise. Bottom-Right: MPM for models identified for 100% noise.

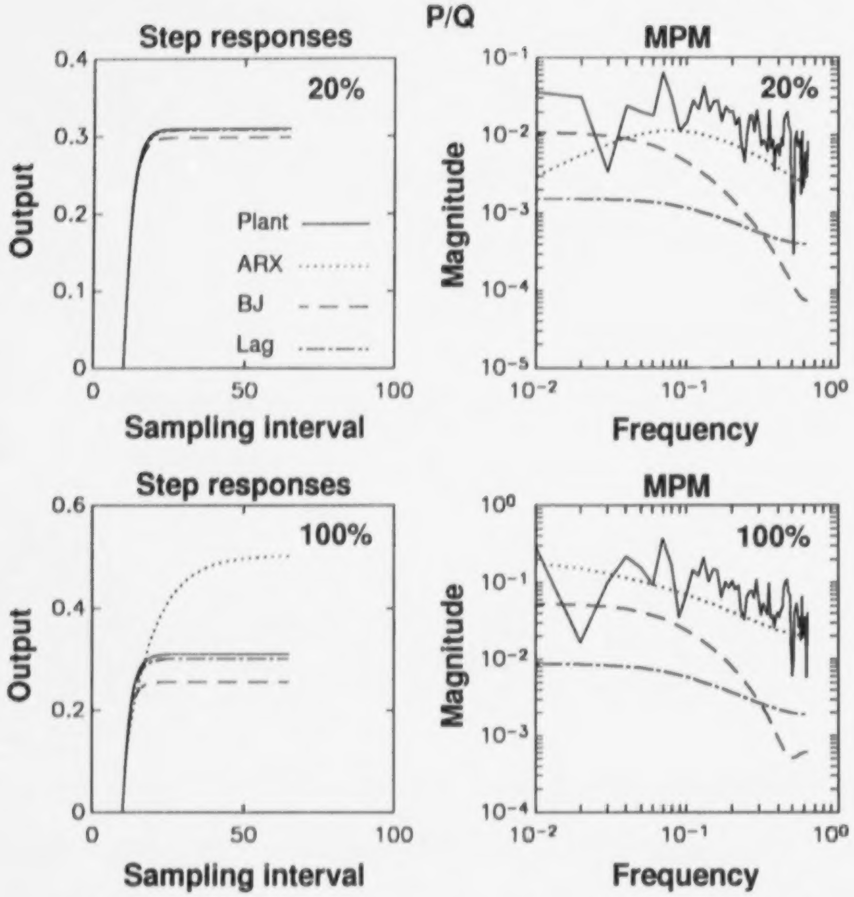


Figure 3.18: Step responses and MPM for different identified models for channel P/Q. Top-Left: Step responses of models identified for 20% noise. Top-Right: Step responses of models identified for 100% noise. Bottom-Left: MPM for models identified for 20% noise. Bottom-Right: MPM for models identified for 100% noise.

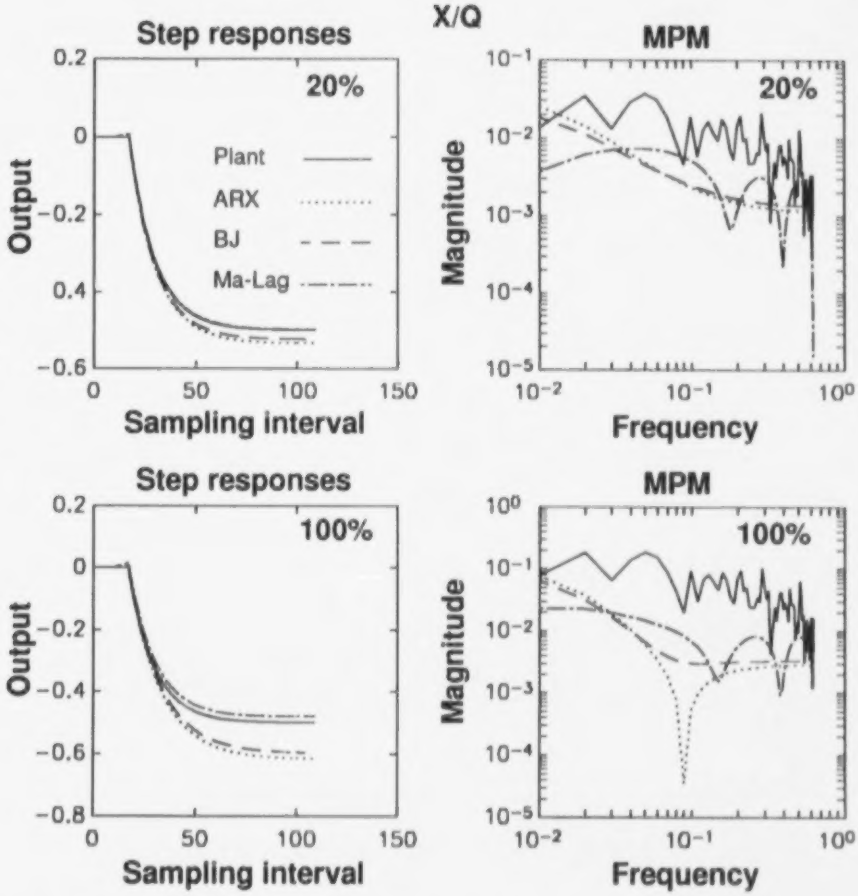


Figure 3.19: Step responses and MPM for different identified models for channel X/Q. Top-Left: Step responses of models identified for 20% noise. Top-Right: Step responses of models identified for 100% noise. Bottom-Left: MPM for models identified for 20% noise. Bottom-Right: MPM for models identified for 100% noise.

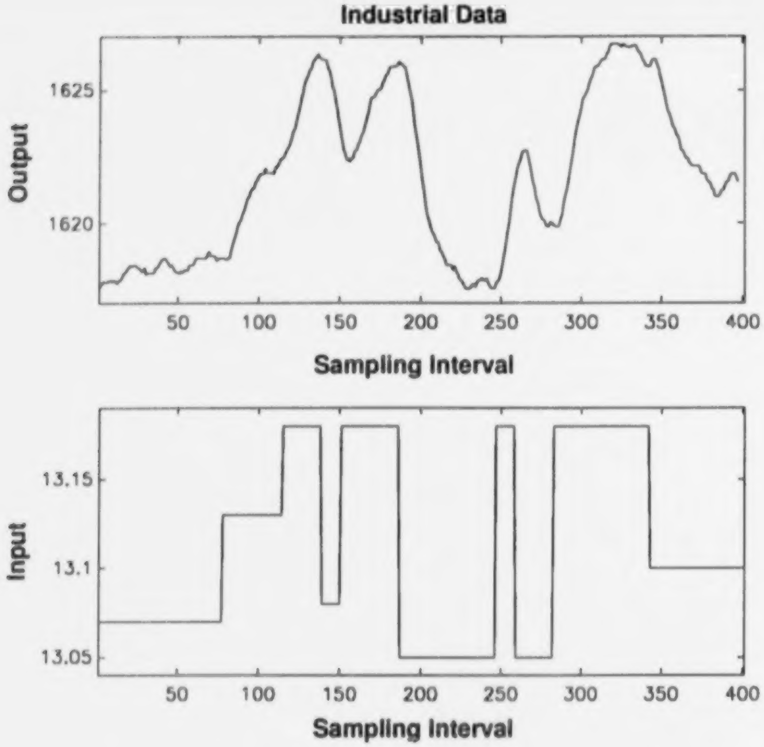


Figure 3.20: Process output and input data.

3.11 Case-II: An industrial plant

The main process is an exothermic catalytic reactor that converts a mixture of superheated NH_3 (at 300°F) and compressed Air to NO_2 (other byproducts are H_2 and NO_x) for $Urea$ production in a fertilizer plant [21]. It is desired to control the temperature $y(t)$ of the catalyst by manipulating the NH_3 flow rate $u(t)$. Figure 3.20 shows a record of openloop I/O data collected at 5 sec. intervals by subjecting the reactor to a series of step inputs. This set of data are designated as $set-1$ and it is used for model identification. For the purpose of validation, two other sets of openloop data are available and they are designated as $set-2$ and $set-3$ in this chapter. The data sets 2 and 3 are shown later in this section.

Most work on the identification of this process had already been carried out by Miller [21]. However this problem is included to show: (a) the use of Markov-Laguerre model to describe the process dynamics; and (b) a transfer function model that gives a better fit in the frequency domain can be obtained. For the sake of brevity, most of the identification work carried by Miller [21] is not repeated here.

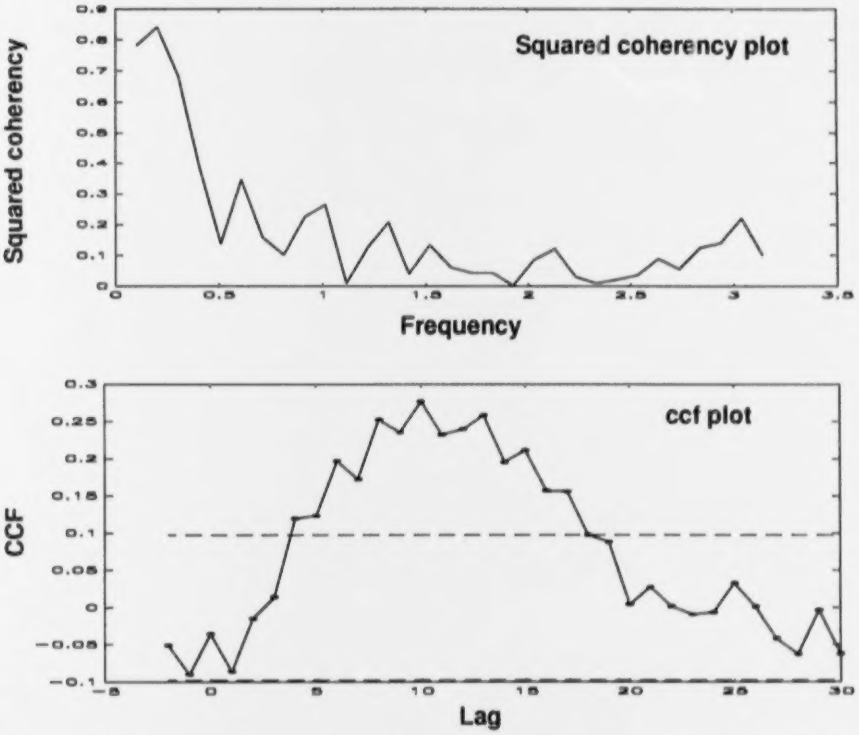


Figure 3.21: Preliminary data analysis. Top: Squared coherency function. Bottom: Cross correlation function.

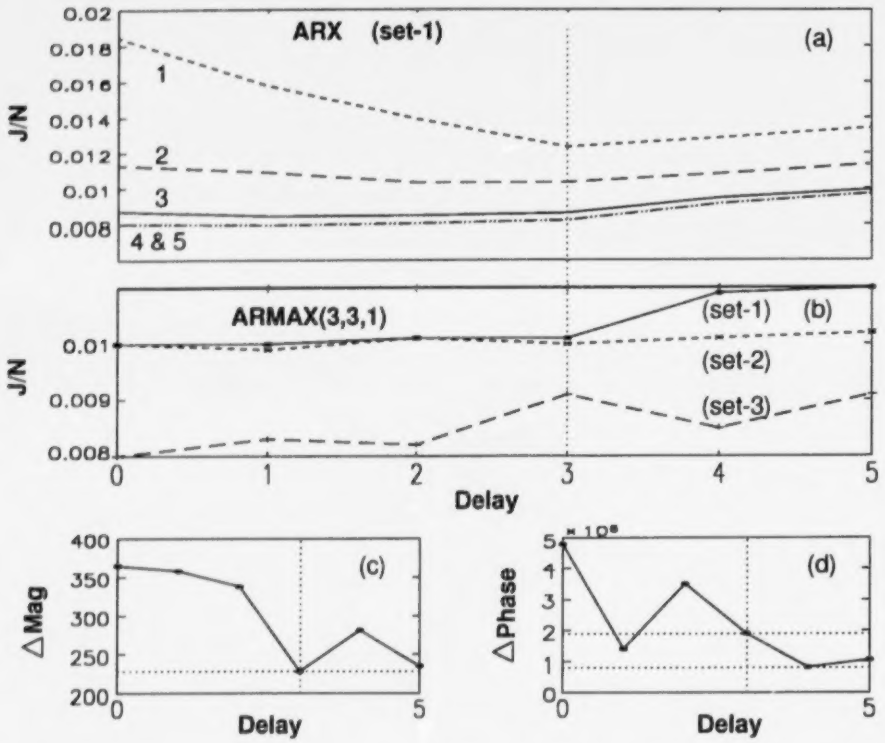


Figure 3.22: Different error analyis. (a) Delay estimation using AUDI for ARX model structure. (b) Effect of delay on the loss functions of ARMAX(3,3,1) model for data set 1, 2 and 3. (c) Effect of delay on $(|G_\phi(\omega)| - |\hat{G}(\omega)|)^2$. (d) Effect of delay on $(\text{ang}G_\phi(\omega) - \text{ang}\hat{G}(\omega))^2$.

3.11.1 Preliminary analysis

Figure 3.21 indicates that the coherency between $y(t)$ and $u(t)$ is weak at the higher frequencies thus implying difficulties may be encountered in identifying models at those frequencies. Not much can be inferred from the CCF plot in Figure 3.21 because the confidence bound is very wide, owning to the small data record. Nevertheless the ccf plot indicates that the process has a delay which is more than 1 and its most likely value is between 2 and 4.

3.11.2 Estimation of models

A preliminary estimate of the model order and delay for an ARX model can be made from the loss function profile obtained using the AUDI method. The variation in loss functions *w.r.t.* the delays for different model orders estimated from data set-1 are shown in Figure 3.22(a). The loss function in Figure 3.22(a) shows that a 3^{rd} order model with 1 delay (excluding ZOH) is adequate to model the system [21]. It is further shown by Miller [21] that a 3^{rd} order ARMAX(na,nb,nc,d) model (where $na = nb = 3$, $nc = 1$) with delay $d = 1$

(excluding ZOH) gives a better representation of the plant dynamics. This model no doubt gives a minimum value for the loss function, but it does not fit too well with the spectral Bode estimates at certain frequencies. It is therefore expected that by changing the delay a better fit can be obtained in the frequency domain.

The effect of delay is explored by plotting the loss function for ARMAX(3,3,1) model for delays from $d = 0$ to $d = 5$ for data sets-1 to 3 as shown in Figure 3.22(b)[15]. Only the loss function for set-1 needs consideration in Figure 3.22(b) because the loss functions for data set-2 and 3 are much smaller. The loss function for set-1 in Figure 3.22(b) shows that this value remains almost constant upto $d = 3$ and after which it starts to increase.

As a next step $\vartheta_M = \sum_{i=1}^{M}(|G_\phi(\omega_i)| - |\hat{G}(\omega_i)|)^2$ and $\vartheta_\phi = \sum_{i=1}^{M}|\angle G_\phi(\omega_i) - \angle \hat{G}(\omega_i)|^2$ are plotted in Figures 3.22(c) and 3.22(d) to examine the effect of delay on the overall Bode performance, where $G_\phi(\omega_i)$ is the model obtained using spectral analysis at the i^{th} frequency ω_i. Figure 3.22(c) shows that the magnitude spectrum of ARMAX(3,3,1,d) fits well with its spectral estimate for $d = 3$ whereas at $d = 4$ the phase spectrum fits the best. Therefore from Figures 3.22(b) to (d) it can be concluded that $d = 3$ is a good compromised candidate rather than $d = 1$ in [21] because:

- The loss function for $d = 3$ is only marginally more than for $d = 1$ and therefore they can be treated as equivalent. Hence from the time domain validation point of view any value of d between 0 and 3 should be acceptable.
- ϑ_M is minimal for $d = 3$; for $d = 4$ this value is slightly higher; and for $d = 0$ to 2, ϑ_M is significantly higher. Therefore the ϑ_M plot indicates that $d = 3$ is most suitable and $d = 4$ may also be accepted as a compromise.
- ϑ_ϕ is minimal for $d = 4$; for $d = 1$ and $d = 3$, the ϑ_ϕ values are almost in the same range; and they are slightly higher than the minimum ϑ_ϕ. ϑ_ϕ values are significantly high for $d = 0$ and $d = 2$. Therefore from the ϑ_ϕ plot it is deduced that $d = 4$ is an ideal choice but $d = 1$ and $d = 3$ should also be acceptable as a compromise.

From the above analysis it is seen that $d = 3$ is arguably the best value in both time and frequency domain validations. Therefore in light of the above analysis, the best ARMAX(3,3,1,3) model is identified (using PEM) to be:

$$\underbrace{(1 - 1.4256q^{-1} + 0.2284q^{-2} + 0.2205q^{-3})}_{A(q)}y(t) =$$
$$\underbrace{(2.1047q^{-1} - 1.0421q^{-2} + 0.4944q^{-3})}_{B(q)}q^{-3}u(t)$$

---

[15] For each delay d, the parameters of ARMAX(3,3,1,d) are identified from set-1 and these parameters are also used to calculate the loss functions for data sets-2 and 3.

$$+\underbrace{(1-0.3811q^{-1})}_{C(q)}\xi(t) \tag{3.41}$$

whose poles are: $\{0.8967, 0.8271, -0.2975\}$ and zeros are: $\{0.2476 \pm j0.4167\}$. Assuming the dominant pole to be at $a = 0.89$ (from equation-(3.41)) a Markov-Laguerre model with 5 Markov and 5 Laguerre coefficients was determined to be (using BLS and set-1):

$$\begin{aligned} \text{Markov Parameters} &\Rightarrow 0.0055, 0.7622, 1.0273, 2.1901, 2.1675 \\ \text{Laguerre Parameters} &\Rightarrow 10.8214, 4.0585, -1.2158, 0.2795, -0.6167 \end{aligned} \tag{3.42}$$

3.11.3 Model validation

Figure 3.23 shows the Bode plots for frequency domain model validation. It is observed from Figure 3.23 that: (a) magnitude plot corresponding to ARMAX(3,3,1,3) is in good agreement with the spectral plot; (b) magnitude spectrum of Markov-Laguerrre model fits better than the ARMAX(3,3,1) models with delays 0 to 2 but not necessarily when compared with the the ARMAX(3,3,1,3) model; and (c) phase plot of ARMAX(3,3,1,3) is in good agreement with the spectral plots.

Figure 3.24 shows that the model outputs $\hat{y}(t)$ for ARMAX(3,3,1,3) and Markov-Laguerre models are in good agreement with the data sets-1 to 3 and they are confined between $\hat{y}(t)$ corresponding to ARMAX(3,3,1,0) and ARMAX(3,3,1,4) models. The feedback output $\hat{y}(t+1|t)$ for the Markov-Laguerre model gives significant variation when compared with the $\hat{y}(t+1|t)$ for ARMAX model because the Markov model translates into a very high order ARX model with $na = 5$ and $nb = 10$.

3.11.4 Residual analysis

The residual analysis for ARMAX(3,3,1,3) model in Figure 3.25 indicates that the filtered residual is white, thus confirming that the assumed noise structure is correct and the model has an EE form. The residual is filtered using $A(q)/C(q)$.

The residual analysis for the Markov-Laguerre model in Figure 3.26 indicates that the filtered residual (filter is: $A(q)/C(q)$) is not quite white but it is close to a white noise signal. Nevertheless, Markov-Laguerre model gives a good fit in the time and frequency domain, therefore no further attempt is made to improve this model.

3.11.5 Step responses

The step response plots for Markov-Laguerre and ARMAX(3,3,1) models for $d = 0, 1, 3$ and 4 are shown in Figure 3.27. It is difficult to say which of these step responses reflect the true noise-free process behavior because the actual plant is unknown. Grossly it can be said that the steady state gain is between 60 and 70 and it has an approximate rise time

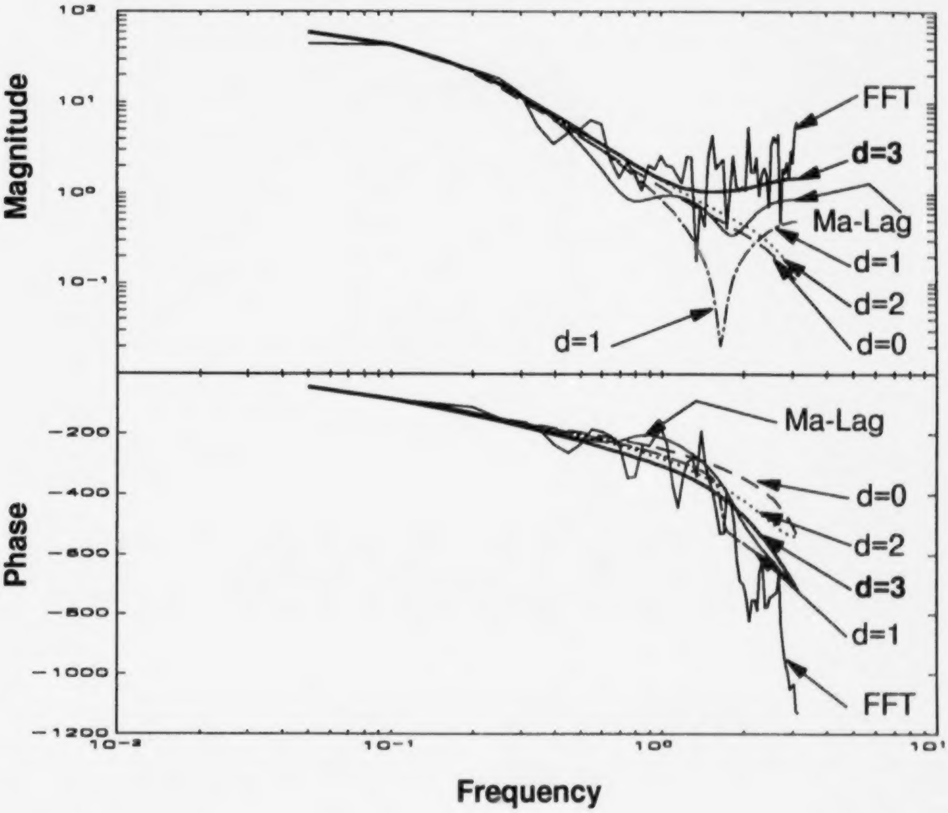


Figure 3.23: Frequency domain model validation. Top: Magnitude spectrum. Bottom: Phase spectrum.

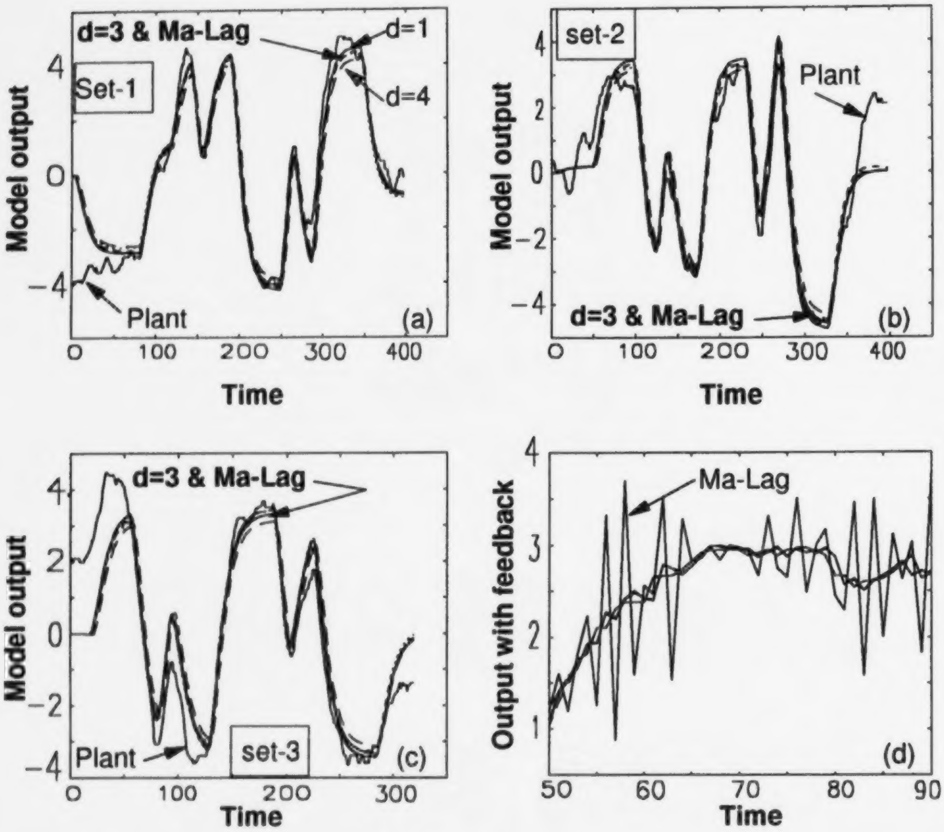


Figure 3.24: Time domain model validation. (a) Model output for data set-1. (b) Model output for data set-2. (c) Model output for data set-3. (d) Model output with feedback for data set-1.

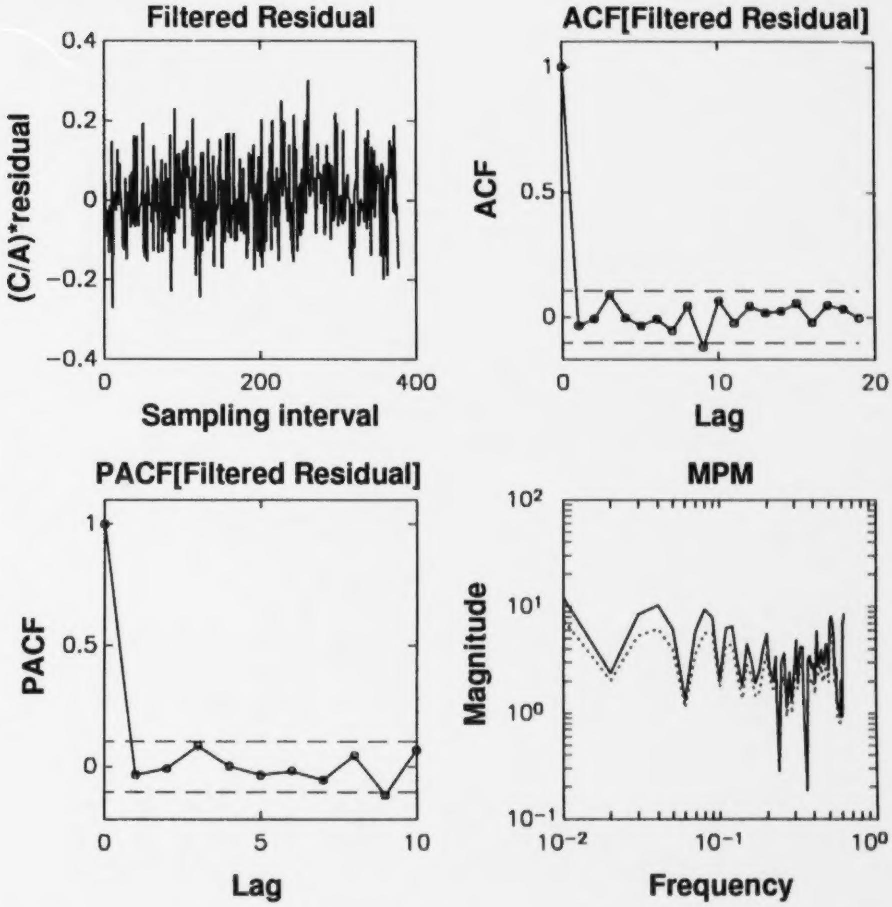


Figure 3.25: Residual analysis for ARMAX model. Top-Left: Filtered residual data. Top-Right: ACF of filtered residual. Bottom-Left: PACF of filtered residual. Bottom-Right: MPM.

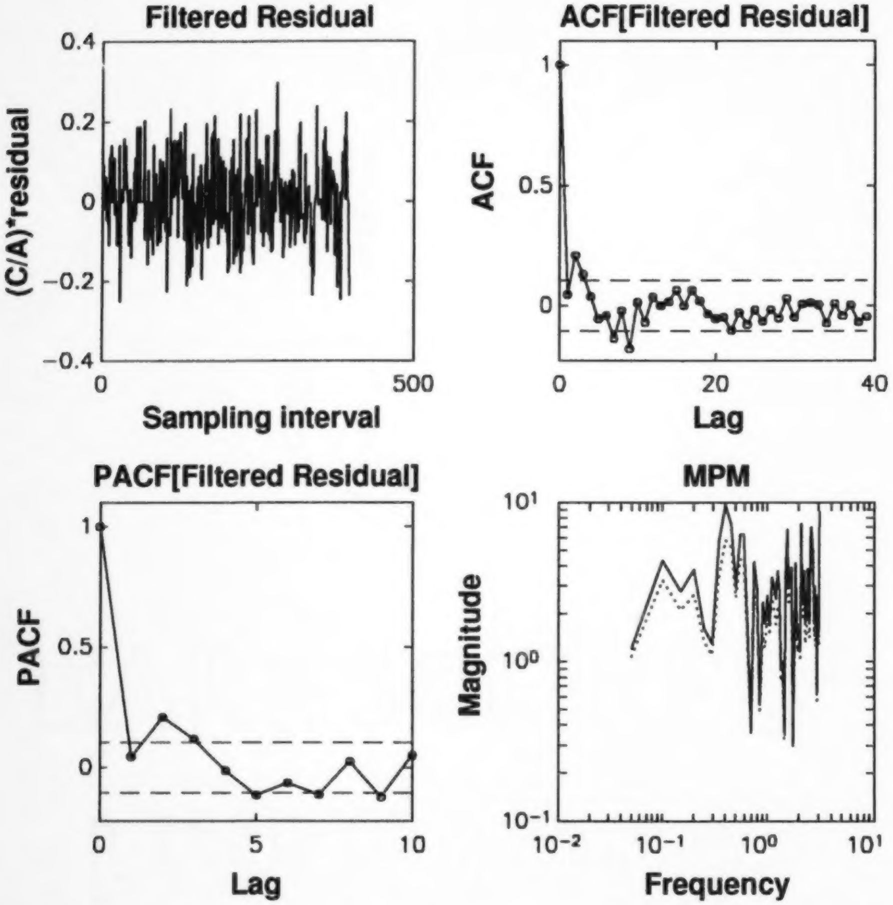


Figure 3.26: Residual analysis for Markov-Laguerre model. Top-Left: Differenced residual data. Top-Right: ACF of differenced residual. Bottom-Left: PACF of differenced residual. Bottom-Right: MPM.

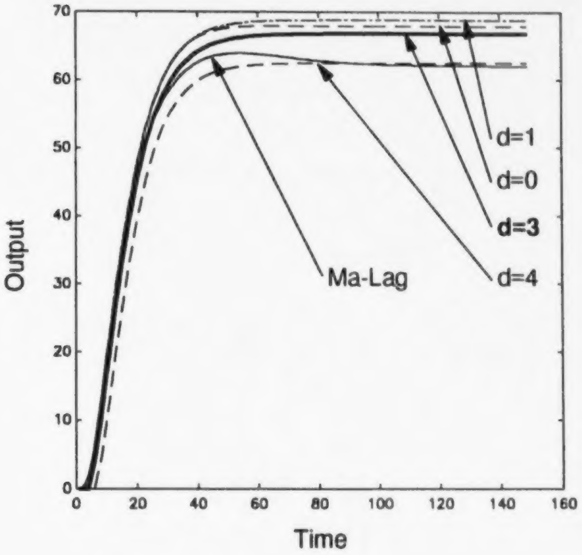


Figure 3.27: Step responses for different identified models.

of 60 × 5 secs. However based on the model validation methods, it can be said that the step response corresponding to the ARMAX(3,3,1,3) model most likely closest to the true behavior of the noise-free plant.

3.12 Conclusions

This chapter summarizes the properties of several I/O models, their estimation methods and highlights their use through the identification of two industrial processes. The models considered in this chapter are: (a) frequency response; (b) FIR/step response; (c) transfer function; and (d) orthonormal function (*i.e.* Markov-Laguerre) models.

A frequency response model is estimated using either signal processing methods discussed in Chapter 2 or other forms of mathematical models can be converted into this form. Step/FIR models are estimated using step tests or by using the BLS method. The parameters of a generalized transfer function model are estimated using the PEM method, whereas BLS or AUDI method can be used to identify simple ARX models. For ARX models, AUDI can: (a) simultaneously estimate parameters/loss functions of different model orders and (b) estimate an appropriate delay for each model order. The dynamics of a linear process can also be expressed by a series of orthonormal functions. Orthonormal function models have OE structure, whereas a generalized transfer function model has an EE structure.

System ID basically comprises of: (a) experiment design and preliminary analysis; (b) estimation of model parameters; and (c) model validation. The purpose of experiment design is to collect appropriate data samples for ID and validation by providing suitable excitation. For preliminary analysis: (a) CCF is used to ascertain system delay; (b) the extent of I/O correlation is checked using squared coherency plots; and (c) the Bode characteristics of the plant are estimated using the signal processing methods discussed in Chapter 2. *A priori* knowledge of the noise model (if available) is used to determine if the use of EE or OE structured model is appropriate. The AUDI method is then used to determine/approximate model order/delay/parameters of ARX models and this information can also be extended to set the order of more generalized transfer function models. The identified models are validated in time/frequency domain and also by using residual analysis.

System ID methods to a great extent are case-specific but they conform to the general ID method outlined in this chapter and this is successfully illustrated via the following two industrial case studies.

(1) *Shell benchmark problem*: The estimation of the noise model helped in designing appropriate filters for process ID. Through ccf and coherency tests it was established that there were no dynamics in the channel X/D. The CCF and spectral Bode (phase) plots revealed that the channels P/D and P/Q were devoid of any delay including ZOH and there was a delay of 7 units (including ZOH) in the channel X/Q. Three types of models namely ARX, BJ and Markov-Laguerre models were considered for system ID. It was observed that with the increase in noise level the parameters of ARX models became more biased when compared with the BJ and Markov-Laguerre models for all the three channels. The estimated models were successfully validated in the time/frequency domain and also through residual analysis.

(2) *An industrial ID problem*: Three sets of data were provided, where one set was used for ID and the other two sets for validation. The squared coherency plot showed poor I/O correlation at the higher frequencies and since the data record was short, ccf could not definitively provide an estimate of the process delay. Loss function analysis using ARX model indicated that a 3^{rd} order model should adequately represent the process dynamics. In order to take the noise dynamics into account an ARMAX(3,3,1) model was selected [21]. After examining the loss function and the error between spectral and model Bode plots for ARMAX(3,3,1) model for different delays it was concluded that a delay of 3 was the best. The resultant ARMAX(3,3,1,3) model was successfully validated in time/frequency domain and also through residual analysis. A Markov-Laguerre model was also identified for this process, but the overall performance of the ARMAX(3,3,1,3) model was better than this model.

Table 3.1: Process objectives, operating conditions and constraints

Variable		Nominal setpoints	Normal operation	Operating objectives
Top pressure	P	2800	$2700 \leq P \leq 2900$	$P < P_{\max} = 3200$
Bottom impurity	X	500	$250 \leq X \leq 1000$	$X < X_{\max}$
Overhead	D	20	$10 \leq D \leq 30$	Minimize D
Reboiler Duty	Q	2500	$2000 \leq Q \leq 3000$	Minimize Q

Where input excitation magnitudes were: $D = \pm 10$ and $Q = \pm 200$

Table 3.2: Estimated models for P/D for different SNRs.

True P/D: $\frac{-0.6096}{1-0.8701q^{-1}}$			
SNR	20%	50%	100%
ARX	$\frac{-0.611+0.0158q^{-1}}{1-0.8774q^{-1}}$	$\frac{-0.6172+0.0538q^{-1}}{1-0.9030q^{-1}}$	$\frac{-0.62+0.0922q^{-1}}{1-0.9285q^{-1}}$
BJ	$\frac{-0.6121+0.0026q^{-1}}{1-0.8674q^{-1}}$	$\frac{-0.6160+0.0080q^{-1}}{1-0.8646q^{-1}}$	$\frac{-0.6219+0.0171q^{-1}}{1-0.8607q^{-1}}$
Ma-Lag	$a = 0.87$ $g_1 = -1.2307$	$a = 0.875$ $g_1 = -1.2329$	$a = 0.875$ $g_1 = -1.2564$

Table 3.3: Estimated models for P/Q for different SNRs.

True P/Q: $\frac{0.1055}{1-0.6597q^{-1}}$			
SNR	20%	50%	100%
ARX	$\frac{0.1041-0.0073q^{-1}}{1-0.6871q^{-1}}$	$\frac{0.1044-0.0277q^{-1}}{1-0.7851q^{-1}}$	$\frac{0.1049-0.0476q^{-1}}{1-0.8774q^{-1}}$
BJ	$\frac{0.1049+0.0002q^{-1}}{1-0.6484q^{-1}}$	$\frac{0.1039+0.0007q^{-1}}{1-0.6292q^{-1}}$	$\frac{0.1024+0.0022q^{-1}}{1-0.5909q^{-1}}$
Ma-Lag	$a = 0.66$ $g_1 = 0.1396$	$a = 0.66$ $g_1 = 0.1383$	$a = 0.66$ $g_1 = 0.1363$

Table 3.4: Estimated models for X/Q for different SNRs.

True X/Q: $\frac{-0.03825q^{-7}}{1-0.9235q^{-1}}$			
SNR	20%	50%	100%
ARX	$\frac{-0.0399q^{-7}+0.0006q^{-8}}{1-0.9265q^{-1}}$	$\frac{-0.0403q^{-7}+0.0015q^{-8}}{1-0.9306q^{-1}}$	$\frac{-0.0410q^{-7}+0.0030q^{-8}}{1-0.9384q^{-1}}$
BJ	$\frac{-0.0401q^{-7}+0.0008q^{-8}}{1-0.9250q^{-1}}$	$\frac{-0.0404q^{-7}+0.0020q^{-8}}{1-0.9286q^{-1}}$	$\frac{-0.0408q^{-7}+0.0042q^{-8}}{1-0.9388q^{-1}}$
Ma-Lag	$a = 0.92$	$a = 0.93$	$a = 0.93$
	$h_1 = 0.0017$	$h_1 = 0.0029$	$h_1 = 0.0048$
	$h_2 = 0.0007$	$h_2 = 0.0006$	$h_2 = 0.0005$
	$h_3 = 0.0008$	$h_3 = 0.0010$	$h_3 = 0.0014$
	$h_4 = 0.0008$	$h_4 = 0.0011$	$h_4 = 0.0016$
	$h_5 = 0.0009$	$h_5 = 0.0014$	$h_5 = 0.0023$
	$h_6 = 0.0015$	$h_6 = 0.0028$	$h_6 = 0.0050$
	$g_1 = -0.0999$	$g_1 = -0.0986$	$g_1 = -0.0963$

Bibliography

[1] T. Söderström and P. Stoica. *System Identification.* Prentice Hall, New Jersey, USA, 1989.

[2] L. Ljung. *System Identification-Theory For The User.* Prentice Hall, New Jersey, USA, 1987.

[3] G.A. Dumont, A. Elnaggar, and A. Elshafei. "Adaptive Predictive Control of Systems with Time-Varying Time Delay". *Int. J. of Adaptive Control and Signal Processing*, 7(6):91 – 101, 1993.

[4] B. Wahlberg. "System Identification Using Laguerre Models". *IEEE Trans. on AC*, 36(5):551 – 562, 1991.

[5] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* The MIT Press, Cambridge, USA, 1986.

[6] B.J. Cott. "(a) Introduction to the Process Identification Workshop at the 1992 Canadian Chemical Engineering Conference. (b) Summary of the Process Identification Workshop at the 1992 Canadian Chemical Engineering Conference". *Journal of Process Control*, 5(2):(a)69–67,(b)109–113, 1995.

[7] P. Banerjee, S.L. Shah, S. Niu, and D.G. Fisher. "Identification of Process Dynamics for Shell Benchmark Problem". *Journal of Process Control*, 5(2):85 – 97, 1995.

[8] D.E. Seborg, T.F. Edgar, and D.A. Mellichamp. *Process Dynamics and Control.* John Wiley & Sons, USA, 1989.

[9] K.J. Åström and B. Wittenmark. *Computer-Controlled Systems.* Prentice Hall, New Jersey, USA, 1990.

[10] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis, Second Edition.* Prentice Hall, New Jersey, USA, 1988.

[11] S. Niu, D.G. Fisher, and D. Xiao. "An Augmented UD Identification Algorithm". *Int. J. Control*, 56(1):193 – 211, 1991.

[12] G.Y. Dumont, C.C. Zevros, and G.L. Pageau. "Laguerre-based Adaptive Control of pH in an Industrial Bleach Plant Extraction Stage". *Automatica*, 26(4):781 – 787, 1990.

[13] J.T. Oden. *Applied Functional Analysis*. Prentice Hall, New Jersey, USA, 1979.

[14] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, USA, 1978.

[15] M. Laszlo. *Hilbert Space methods in Science ad Engineering*. Adam Higler, Bristol, U.K. and Akademiai Kiado, Budapest, Hungary., 1989.

[16] Y. Fu and G.A. Dumont. "An Optimal Time Scale for Discrete Laguerre Network". *IEEE Trans. on AC*, 38(6):934 – 938, 1993.

[17] D.J. Cloud and B. Kouvaritakis. "Statistical bounds on multivariable frequency response: an extension of the generalized Nyquist criterion". *IEE proceed. pt-D*, 133(3):97–110, 1986.

[18] G.M. Jenkins and D.G. Watts. *Spectral Analysis and its Application*. Holden-Day, San Franscisco, USA, 1968.

[19] J.D. Cryer. *Time Series Analysis*. PWS-KENT Publishing Company, Boston, USA, 1986.

[20] D.E. Rivera, S.V. Gaikwad, and X. Chen. "Control-ID: A Demonstration Prototype for Control-Relevant Identification". In *American Control Conference*, Baltimore, USA, 1994.

[21] R.M. Miller. *Adaptive Predictive PID*. M.Sc. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada, 1995.

Chapter 4

Parameter Estimation for Orthonormal Function Models using AUDI

The augmented UD identification (AUDI) method is used to simultaneously estimate parameters of all 1 to N^{th} order discrete orthonormal function models in one computational step. This method is tested on different types of orthonormal functions such as the Laguerre, Kautz, FIR and Markov-Laguerre models.

4.1 Introduction

Orthonormal functions are used to represent a function in an expanded form by a series of (orthonormal) functions. There are several types of orthonormal functions, for example, the Fourier series discussed in Chapter 2 is a case in point. Other orthonormal functions *i.e.* Laguerre and Markov-Laguerre functions were briefly introduced in Chapter 3.

The orthonormal functions were originally formulated in the eighteenth and nineteenth centuries and they were in the form of differential equations. However for many engineering applications, some of these orthonormal functions are now used in the form of transfer functions [1, 2, 3, 4]. This thesis also pursues the application of orthonormal functions in the form of transfer functions because these transfer function forms are more relevant to the controller design addressed in this thesis. Nevertheless in order to provide a historical perspective, orthonormal functions in the form of differential equations are briefly discussed in Section 4.2.

---

[1] A version of this chapter has been accepted for publication in: *IEEE Transactions on Signal Processing* as 'Simultaneous Estimation of Parameters for different Orders of Discrete Orthonormal Function Models', by P.Banerjee and S.L.Shah.

Orthonormal functions have been extensively applied to a wide range of areas; some of which were outlined in Chapters 2 and 3. The use of orthonormal functions in numerical analysis, solution of partial differential equations and probability theory are well established [5]. Specifically, orthonormal functions have been proved to be useful to solve: (a) problems in numerial analysis such as numerical integration and interpolation, (b) problems in partial differental equations relevant to quantum mechanics, hydrodynamics, electrostatics, gravitation etc. and (c) problems in probabilities such as two dimensional probability distribution, non-Gaussian distribution and scattering of waves by rough surfaces (useful in radio/radar engineering, acoustics, and optics) [5]. This thesis is confined to using orthonormal functions to represent the plant dynamics as illustrated in Chapter 2 and 3.

Orthonormal functions in the form of transfer function models such as the Laguerre and Kautz function models are based on some *a priori* process information such as the knowledge of dominant poles; hence such models are also called grey-box or semi-empirical models as mentioned in Chapter 3. The need for *a priori* knowledge of the dominant pole of a system by orthonormal functions may be considered to be a disadvantage, but there are several other advantages in using orthonormal function models which outweigh this disadvantage. Moreover in most situations this dominant pole can be approximated either through step tests as suggested by Fu *et al.* [6] and Cluett *et al.* [7, 8] or from ARX models as shown in Chapter 3. Some of the advantages of orthonormal functions have been summarized by Wahlberg [9] and Dumont *et al.* [10] as: (a) insensitivity to sampling time; (b) compact and unstructured model representation, (c) well-conditioned covariance matrix, therefore a numerically robust estimation procedure and (d) a significantly insensitive model estimate to system noise.

Depending on the dynamics of the process, different types of orthonormal functions are used to model them. For example, the Kautz model is useful in approximating the dynamics of an underdamped process as it is based on the dominant complex poles of the system. The use of Kautz function model has been illustrated by Lindskog *et al.* to compactly model a highly resonating aircraft flight flutter system [11].

An important sub-class of the Kautz models are the Laguerre models [9, 6]. A Laguerre model based on the real dominant pole is useful in describing the dynamics of an overdamped plant [9, 12, 7]. Dumont *et al.* [13, 12] and Cluett *et al.* [7] have illustrated the use of Laguerre model via several process control applications. Fu *et al.* have also shown that a Laguerre function with complex poles can be formulated to approximate the dynamics of an underdamped plant [6]. Some other applications of the orthonormal function models include compact data representation [1, 2, 5], frequency smoothing [14] and filter design [3].

The most elementary form of the orthonormal function is the FIR (finite impulse response) model. For example, both Kautz and the Laguerre models can be reduced to the FIR form. The distinction between the orthonormal and black-box models fade out for the FIR structure, because no *a priori* process information is embedded in the FIR model. Infact the black-box step-response model is only a cumulative form of the FIR model. Since Kautz or Laguerre functions have a longer memory, they yield more compact models in comparison to the FIR or step-response models [9]. FIR models are best suited for processses with significant delay, inverse response or other unusual high frequency dynamics, although it requires a large number of coefficients to model them.

A good trade-off between capturing unusual high frequency dynamics and number of model parameters can be achieved by combining the FIR and the Laguerre or Kautz models to form the Markov-Laguerre or Markov-Kautz function models as illustrated by Finn *et al.* via a process control application [15]. The FIR component in this combined model helps to capture the unusual high frequency dynamics whereas the Laguerre or Kautz models can be used to effectively emulate the low frequency dynamics.

The least-squares (LS) technique is a standard system identification tool to estimate the parameters of a specified order of the linear models as described in Chapter 3. Often due to the practical constraints such as the need for lower order controllers, lower order models that satisfactorily approximate the plant dynamics are preferred over the higher order models. Several LS trial runs may be required for different model orders to obtain an acceptable model structure. Therefore the question arises: can a method be devised that can simultaneously estimate the parameters of different orders of the model? Niu *et al.* [16, 17] have addressed this issue by proposing an augmented **UD** identification (AUDI) method, that simultaneously estimates the parameters for all 1 to N^{th} order ARMAX models with approximately the same computational effort as for the N^{th} order ARMAX-LS solution. Therefore the AUDI-ARMAX method solves the LS solutions concurrently for different orders of the ARMAX model. The AUDI method proposed by Niu *et al.* was motivated from the LU-decomposition and Bierman's UD factorization [18, 19]. The LU or UD factorization methods were in turn motivated by numerical considerations in the LS solution. Niu *et al.* have also shown that the AUDI-ARMAX method retains the advantages of numerical stability as provided by the LU or UD decomposition methods [16, 17].

The AUDI-ARMAX solution is based on the decomposition of a covariance matrix that has a symmetrical structure. The approach proposed in this chapter uses the AUDI concepts to ***simultaneously*** estimate the parameters of ***all*** 1 to N^{th} order discrete orthonormal models in one computational step by forming a symmetrical covariance matrix. For output error models, Niu *et al.* have recently also proposed LU decomposition of a covariance matrix that has a non-symmetric structure to simultaneously obtain the model parameters [20].

This chapter first presents a brief historical sketch of the orthonormal functions in Section 4.2 which is followed by a list of some definitions relevant to the orthonormal function models in Section 4.3. The focus of this chapter then shifts to the main theme *i.e.* the AUDI formulation for orthonormal function models in Section 4.4. The application of

AUDI technique to estimate parameters of different orders of orthonormal function models are then discussed in Sections 4.5 to 4.9. Specifically; Sections 4.5 and 4.6 referes to Laguerre models with real and complex poles and Sections 4.7, 4.8 and 4.9 pertains to the Kautz, FIR (and step-response) and Markov-Laguerre models respectively. The performances of the proposed method for various model types are summarized in Section 4.10.

4.2 Historical perspective

Different classical orthonormal functions that were defined in the eighteenth and nineteenth centuries can be arranged in the following chronological order (Table 4.1) [5]:

Table 4.1: Classical Orthonormal Functions and their relevant intervals.

Year	Name	Interval
1785	Legendre	-1,1
1859	Jacobi	-1,1
1859	Chebyshev	-1,1
1864	Hermite	$-\infty,\infty$
1877	Gegenbauer	-1,1
1879	Laguerre	$0,\infty$

Prior to World War II, these classical orthonormal models were considered to be a complicated system of differential equations and for each type of function, it's properties were derived separately. Later in 1948 an Italian mathematician F.G.Tricomi and Russian mathematicians M.A.Lavrentyev and B.V.Shabat in 1951, independently showed that these classical orthonormal functions were special cases of a general weighting function [5][2]. The work by Tricomi and Lavrentyev *et al.* were motivated by the discovery by J.Rodriguez in 1814 who showed that Legendre polynomial (which was the only known polynomial then) could be generated using [5]:

$$P_n(x) = \frac{(-1)^n}{2^n n!}\frac{d^n}{dx^n}(1-x^2)^n \tag{4.1}$$

Tricomi and Lavrentyev *et al.* generalized the above equation-(4.1) to [5]

$$P_n(x) = A_n \frac{1}{w(x)}\frac{d^n}{dx^n}[w(x)\beta^n(x)] \tag{4.2}$$

---

[2] F.G.Tricomi's work was known in 1970 when his book *Vorlesungen über Orthogonalreihen* was published by Springer Verlag in German language. The original paper by Tricomi was published in 1943 in Italian language. Some results published by Lavrentyev *et al.* were obtained to them through private communication from other Russian mathematicians I.G.Aramonovich and N.I.Kozhevnikov. [5].

to represent all classes of orthonormal function models and it was aptly named as the *generalized Rodriguez formula* for classical orthonormal functions. The functions A_n, $w(x)$ and $\beta(x)$ in equation-(4.2) vary with the type of orthogonal models. For example the Laguerre function can be expressed as [5]:

$$l_n^{\lambda}(x) = \overbrace{\frac{1}{n!}}^{A_n} \underbrace{x^{-\lambda}e^{x}}_{w^{-1}(x)} \frac{d^n}{dx^n}(\overbrace{e^{-x}x^{\lambda}}^{w(x)} \underbrace{x^n}_{\beta^n(x)}) \tag{4.3}$$

Similarly other functions can be defined but they are not shown here as they are not relevant to this thesis[3]. It is however interesting to note that: (a) Legendre and Chebyshev functions are special cases of Gegenbauer function and Gegenbauer function in turn is a special case of Jacobi function and (b) Hermite functions can be expressed as Laguerre function [5]. This led Tricomi to conclude that the orthonormal functions can virtually be classified as *i.e.* Jacobi and Laguerre functions [5].

Various orthonormal functions have found applications in different areas such as: (a) Legendre polynomials are useful to solve partial differential equations such as Laplace, Poisson and wave equations in the spherical coordinate, (b) Chebyshev functions are used in filter design, interpolations in numerical methods and in probability theory, (c) Laguerre functions find use in quantum mechanics, systems theory and probability theory etc. and (d) classical Schrödinger's wave equation in quantum mechanics[4] have been solved using Hermite polynomials and they are also used in probalility theory [5].

The orthonormal functions referred in this thesis are confined to modelling dynamic systems (also shown in Chapter 3) and applying them to design predictive controllers which is persued in Chapter 6. Besides the Laguerre functions, some other orthonormal functions that are used in process control are the Meixner, Kautz and Markov functions[5]. In 1934 J.Meixner function proposed Meixner function and W.H.Kautz proposed Kautz function in 1954 [4]. Since Markov function is shown to be a special case of orthonormal functions, this function *i.e.* FIR model is also called Markov (orthonormal) function.

The emphasis of this thesis is on using transfer function models in the discrete domain, therefore the orthonormal functions that are originally in the form of differential equations are converted into (discrete) transfer function forms in the frequency domain. It so happens that Laguerre and Kautz functions can be converted into discrete transfer function forms, hence these functions are mainly referred in this thesis. Meixner functions is not referred in this thesis because it does not have any rational transfer function form in the z-domain [4]. The discrete transfer function forms for the Laguerre and Kautz functions were respectively derived by M.J.Gottlieb in 1938 and P.W.Broome in 1965 [4]. Traditionally Laguerre models are used to describe overdamped systems and Kautz models for underdamped plants; hence Laguerre models can be shown be a special case of Kautz models [9, 6, 4].

---

[3] Any standard handbook on mathematics will have expressions for the orthonormal functions in the form of differential equations.

[4] Quantum mechanics was briefly discussed in Chapter 2.

[5] One of the main reasons why Laguerre, Kautz, Meixner and Markov functions are used in systems engineering because they are applicable in the region 0 to ∞.

4.3 Definitions

Before formally defining the terms 'orthogonal' and 'orthonormal' functions, it is required to go through a series of other definitions that are connected to these orthogonal functions. The following premises from functional analysis are therefore used to formally define the scope of orthonormal functions [21, 22, 23]:

Definition 4.1 Intervals*: An* open interval *is denoted by* $(a, b) = \{x : a < x < b, x \in \Re\}$*.* *A* closed interval *is given by* $[a, b] = \{x : a \leq x \leq b, x \in \Re\}$*. A* half open *(or* half closed*) intervals are similarly defined as* $(a, b]$ *or* $[a, b)$*.*

Definition 4.2 Euclidean and complex spaces*: N-dimensioned* Euclidean *and* complex *spaces are designated as* $\Re^N$ *and* $\Im^N$ *respectively. The underlying vector for* $\Re^N$ *and* $\Im^N$ *comprises of a set of N real or complex numbers (also called as N-tuples). For example,* $x \in \Re^3 \Rightarrow x = [x_1, x_2, x_3]$*.*

Definition 4.3 Metric space, metric*: A* metric space *is a pair* (X, d)*, where* $X \in \Re^N$ *is a nonempty set and d is* metric *(or* distance function*) on* X*. For example, if* $X, Y \in \Re^N$*, then* $d(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$*, where* $x_i \in X$ *and* $y_i \in Y$*.*

Definition 4.4 l^p space and normed[6] l^p space*: Metric induced under* l^p *space is given by* $d_p(x, y) = (\sum_{i=1}^{\infty} |x_i - y_i|^p)^{1/p}$ *where* $x_i \in X$*,* $y_i \in Y$ *and* $p \geq 1$ *is a fixed real number.* l^p *norm of* $x \in \Re^N$ *is given by* $||x||_p = (\sum_{i=1}^{N} |x_i|^p)^{1/p}$*, such that* $\sum_{i=1}^{\infty} |x_i|^p < \infty$*. For real sequences this space is called* real space l^p *and similarly it is called* complex space l^p *for complex sequences.*

Definition 4.5 Function spaces L[a,b]*:* $L[a, b]$ *is a metric space comprising of real valued functions* $x(t) \in X$ *and* $y(t) \in Y$ *which are continuous in* $t \in \Re$ *on* $[a, b]$ *and whose metric is defined as* $d(x, y) = sup_{t \in [a,b]} |x(t) - y(t)|$*. For* $L_p[a, b]$ *space, the metric* d_p *is* $d_p(x, y) = \{\int_a^b |x(t) - y(t)|^p dt\}^{1/p}$*.*

Definition 4.6 Square-Lebesgue integral*: function* $X(t) \in [0, \infty]$ *is* square-Lebesgue *integrable if:*

$$\int_0^{\infty} ||X(t)||^2 dt < \infty \tag{4.4}$$

---

[6] Normed spaces were independently proposed by S.Banach, H.Hahn and N.Wiener all in 1922 [22].

Definition 4.7 Convergence*: A sequence* $\{x_i\} \in X$ *in a metric space* (X, d) *is said to* converge *or to* be convergent *if* $\exists\, x_o : \lim_{i \to \infty} d(x_i, x_o) = 0$*. In such a case* x_o *is called the* limit *of* $\{x_i\}$ *or simply* $x_i \to x_o$*.*

Definition 4.8 Cauchy sequence*: A sequence* $\{x_i\}$ *in* (X, d) *is said to be a* Cauchy sequence *if* $d(x_i, x_j) < \epsilon$ *for all* $i, j > N$*. Every* convergent *sequence in* (X, d) *is a Cauchy sequence.*

Definition 4.9 Complete space*: A space* (X, d) *is said to be* complete *if every Cauchy sequence in* (X, d) *converges.*

Definition 4.10 Banach space*: A complete normed space is called a* Banach space*.* l^p *and* $L_p[a, b]$ *spaces are Banach spaces.*

Definition 4.11 Inner product space*: Inner product space between vectors* $x = [x_1, \ldots, x_n] \in \Im^n$ *and* $y = [y_1, \ldots, y_n] \in \Im^n$ *is given by* $< x, y^H > = x_1 y_1^H + \cdots + x_n y_n^H$*. Similarly inner product on* X *is* $\|x\|_2 = \sqrt{< x, x^H >}$ *and for functions it is* $< x(t), y(t) >= \int_a^b x(t) y^H(t) dt$*. (The superscript* ^{H} *denotes complex conjugate transpose.)*

Definition 4.12 Hilbert space*: A complete inner product space is called a* Hilbert space $\mathcal{H}$*. Every finite dimensional inner product space is complete, therefore they belong to Hilbert space* $\mathcal{H}$*. Obviously Hilbert space is a Banach space, however the converse is not true.* l^p *and* $L_p[a, b]$ *spaces with* $p = 2$ *belong to Hilbert space.*

Definition 4.13 Orthogonal*: Two vectors* $x, y \in \Im^n$ *are said to be orthogonal if* $< x, y^H >= 0$*. Obviously orthogonal vectors belong to* $\mathcal{H}$*.*

Definition 4.14 Orthonormal*: Two orthogonal vectors* $x, y \in \Im^n$ *are orthonormal if* $< x, y^H >= 1$ *for* $x = y$*. Orthonormality condition is compactly denoted by a* Kronecker delta function δ_{mn} *i.e. if* $m = n$, $\delta = 1$ *and* $\delta = 0$ *for* $m \neq n$*.*

The above set of definitions are next used to formally represent the orthonormal function model as follows:

If $y \in L_2[0, \infty)$ (*i.e. square-Lebesgue integrable*) and if there exists *orthonormal* sequences $\{\psi_i\}$ which is *complete*, then $y \in L_2[0, \infty)$ can be expanded into the series in the *Hilbert* space as:

$$y = \sum_{i=0}^{\infty} < y, \psi_i > \psi_i \tag{4.5}$$

The following theorem then applies to equation-(4.5):

Theorem 4.1 Fischer-Riesz Theorem[7]*: Let* $\{\psi_i\}$ *be a complete orthonormal set as defined above and let* $\{g_i\}$ *be a sequence of constants such that* $\sum_{i=0}^{\infty} |g_i|^2$ *is convergent, then the series*

---

[7] This theorem was proved by E.Fischer and F.Riesz independently in 1907 [24]

$$y = \sum_{i=0}^{\infty} g_i \psi_i \tag{4.6}$$

converges to y whose coefficients g_i are given by $g_i = < y, \psi_i >$ [24, 25].
Proof: See[8] Theorem-20 in page 20 in reference [24].
The series given by equation-(4.6) is also called as a *Fourier series* where g_i is the i^{th} *Fourier coefficients* of y with respect to ψ_i [24]. Orthonormal vectors ψ_i in equation-(4.6) are called as *orthonormal basis* in $\mathcal{H}$. The convergence of Orthonormal series equation-(4.6) can be defined as:

Definition 4.15 Convergence of orthonormal series: *The orthonormal series (or Fourier series)* $y = \sum_{i=0}^{\infty} g_i \psi_i$ *is convergent only if the series* $\sum_{i=0}^{\infty} |g_k|^2$ *converges.*

If y in equation-(4.6) be the system output as described by Figure 2.1 (in Chapter 2), then the impulse response for such a system is L_2 stable. The system output y being a function of time t, it is designated as $y(t)$, similarly ψ is written as $\psi(t)$. Processes such as pure integrators, oscillators and unstable systems cannot be expressed by equation-(4.6) as they do not lie in the L_2 space [25].

When the orthonormal basis signal $\psi_i(t)$ in equation-(4.6) is a function of time, then $\psi_i(t)$ can be expressed as a set of differential equations as given by equation-(4.3). However for system identification (ID)/process control applications it is more convenient to use transfer function forms. Therefore let $\Psi_i(z)$ denote the discrete domain transfer function of $\psi_i(t)$, then the system dynamics is expressed as:

$$G(z) = \sum_{i=0}^{\infty} g_i \Psi_i(z) \tag{4.7}$$

The basis function in $\Psi_i(z)$ in equation (4.7) is chosen such that it reflects the unit delay (due to zero order hold ZOH) that is implicit in the discrete transfer function $G(z)$ and obviously $\Psi_i(z)$ hold the orthonormality condition:

$$\langle \Psi_i(z), \Psi_j(z) \rangle = \delta_{i,j} \tag{4.8}$$

The innerproduct in equation (4.8) is between the impulse responses of $\Psi_i(z)$ and $\Psi_j(z)$. With reference to the note:

Note 4.1 *Applying* Remark-2.1 *and* Note-3.1, *the basis signal $\psi_i(t)$ for $\Psi_i(z)$ can be expressed as: $\psi_i(t) = \Psi_i(q)u(t)$, for the referred system given by Figure 2.1 in Chapter 2.*

---

[8] The proof is not included because: (a) for brevity, (b) it is straight forward and (c) the application of this theorem is only of interest in this thesis.

the true plant output is expressed as:

$$y(t) = \sum_{i=0}^{\infty} g_i \Psi_i(q) u(t) \tag{4.9}$$

In practice only a finite number of orthonormal functions are required to approximately represent the process dynamics. Therefore the model output for an N^{th} order orthonormal series is given by:

$$\hat{y}(t) = \sum_{i=1}^{N} g_i \Psi_i(q) u(t) \tag{4.10}$$

For a finite order orthonormal function, the following definitions hold:

Definition 4.16 Maximal set*: Orthonormal set $B = \{\psi_i\}$ is* maximal *if* $\nexists x_0 \in \mathcal{H}$ *such that* $\{x_0 \notin B\}$ *and* $\{x_0 \cup B\}$ *is an orthonormal set. A maximal set is also called a* complete set[9].

For a complete set, equation-(4.9) and the following definition holds.

Definition 4.17 Parseval's Theorem[10]*: For a maximal orthonormal set $\{\psi_i\}$ the* Parseval's theorem *is given by:* $||y||^2_{\mathcal{H}} =< y, y >_{\mathcal{H}} =< \sum_{i=1}^{\infty} g_i\psi_i, \sum_{j=1}^{\infty} g_j\psi_j >_{\mathcal{H}} = \sum_{i=1}^{\infty} |g_i|^2$.

in other words, Parsevals's relation can also be stated as:

$$\sum_{i=1}^{t} y^2(i) = \sum_{k=1}^{N}\sum_{i=1}^{t} [g_k(t)\psi_k(i)]^2 \tag{4.11}$$

Definition 4.18 Bessel's inequality*: If $\{\psi_i\}$ is not maximal then* Bessel's ineqality *is given by:*

$$\begin{aligned} ||y - \sum_{i=0}^{N} g_i\psi_i|| &< \epsilon \\ &\textit{alternately} \\ ||\sum_{i=0}^{N} g_i\psi_i|| &< ||y|| \end{aligned} \tag{4.12}$$

where $\epsilon > 0$ and N is a positive integer. In practice most models are finite order, therefore they obey Bessels inequality. The model given by equation-(4.10) can also be written as:

---

[9] Definition-4.9 is not to be confused with definition-4.16.

[10] Parseval's theorem can be stated in many ways. One form of Parseval's equation was shown in Chapter 2 and in Chapter 7 this theorem is again re-visited.

$$\hat{y}(t) = \phi^T(t)\theta \tag{4.13}$$

where $\phi(t) = [\phi_1(t), \phi_2(t), \ldots, \phi_N(t)]^T$ is the data vector and $\theta = [g_1, g_2, \ldots, g_N]^T$ is the parameter vector and where $\phi_i(t) = \Psi_i(q)u(t)$. The LS parameter estimates $\hat{\theta}$ of θ in equation-(4.13) is given by:

$$\hat{\theta} = [\Phi^T\Phi]^{-1}\Phi^T Y \tag{4.14}$$

where $Y = [y(1), y(2), \ldots, y(N)]^T$ is the output vector and $\Phi = [\phi(1), \phi(2), \ldots, \phi(N)]$ is the data matrix.

Definition 4.19 *Let the LS solution given by equation-(4.14) be designated as the LS_y solution.*

Orthonormal models give very robust parameter estimates for plants having OE (output error) structure as shown by an illustration for the following OE structured plant:

$$y(t) = \underbrace{(\hat{G}(q) + \tilde{G}(q))}_{G(q)} u(t) + e(t) \tag{4.15}$$

where the plant $G(q)$ in equation-(4.15) is expressed as a summation of the nominal model $\hat{G}(q)$ and its associated unmodelled dynamics $\tilde{G}(q)$; and $e(t)$ is either a white or colored noise sequence. Letting:

$$\begin{aligned} \hat{G}(q) &= \sum_{i=1}^{N} g_i \Psi_i(q) \\ \tilde{G}(q) &= \sum_{i=N+1}^{\infty} g_i \Psi_i(q) \end{aligned} \tag{4.16}$$

then combining equations-(4.13), (4.15) and (4.17) yield:

$$Y = \Phi^T\hat{\theta} + \tilde{\Phi}^T\tilde{\theta} + E \tag{4.17}$$

where $E = [e(1), e(2), \ldots, e(N)]$, $\tilde{\Phi}$ and $\tilde{\theta}$ are the data matrix and parameter vector corresponding to $\tilde{G}(q)$. Substituting Y from equation-(4.17) into equation-(4.14) and then taking its expectation ($\mathcal{E}$) gives:

$$\begin{aligned} \mathcal{E}[\hat{\theta}] &= \theta + \mathcal{E}[[\Phi^T\Phi]^{-1}\Phi^T\tilde{\Phi}^T\tilde{\theta}] + \mathcal{E}[[\Phi^T\Phi]^{-1}\Phi^T E] \\ &= \theta \end{aligned} \tag{4.18}$$

The above conclusion holds only if: (a) $u \in \mathcal{N}(\sigma^2, 0)$ and (b) E is zero mean; implying Φ and E are uncorrelated. And that also implies $\boldsymbol{\Phi}$ and $\tilde{\boldsymbol{\Phi}}$ are uncorrelated.

4.4 AUDI formulation of orthonormal function models

This section discusses the use of AUDI method to simultaneously estimate the parameters of different orders of the orthonormal function models. Because of the simultaneous nature of the model estimation, it is required to associate the model outputs and parameters with the model order. For example the model output corresponding to the N^{th} order model is given by:

$$\hat{y}(t,N) = \sum_{k=1}^{N} g_k \psi_k(t) \tag{4.19}$$

where the N^{th} order model is: $\hat{G}(z,N) = \sum_{k=1}^{N} g_k \Psi_k(z)$. Parameters associated with different model orders of the model are given in the next subsection.

4.4.1 Data regressor and parameters for the AUDI method

The first step in extending the AUDI method [16, 17] to simultaneously estimate the parameters of all 1 to N^{th} order orthonormal filter models is to formulate an appropriate data regressor. To begin with, equation (4.19) is re-expressed as:

$$\hat{y}(t,N) = h^T(t,N)\theta(t,N) \tag{4.20}$$

where the data vector $h(t,N)$ is given by:

$$h(t,N) = [\psi_1(t), \psi_2(t), \ldots, \psi_N(t)]^T \tag{4.21}$$

and the parameter vector, $\theta(t,N)$ is:

$$\theta(t,N) = [g_{1N}(t), g_{2N}(t), \ldots, g_{NN}(t)]^T \tag{4.22}$$

where $g_{kN}(t), k = 1 \ldots N$ in equation (4.22) are parameters corresponding to the model order N. The parameter g_k in equation (4.19) is re-written as g_{kN} in (4.22) to signify that it is the k^{th} parameter corresponding to the N^{th} order model. As per the AUDI formulation, the data vector:

$$\phi(t,N) = [-y(t), \psi_1(t), \ldots, \psi_N(t)]^T = \left[-y(t), h^T(t,N)\right]^T \tag{4.23}$$

is defined corresponding to the model:

$$-y(t) + g_{1N}(t)\psi_1(t) + g_{2N}(t)\psi_2(t) + \cdots + g_{NN}(t)\psi_N(t) = 0 \tag{4.24}$$

to estimate the parameter matrix (or the LS_y model-set) $\Theta(t,N)$:

$$\begin{aligned}\Theta(t,N) &= [\theta(t,1),\theta(t,2),\ldots,\theta(t,N)] \\ &= \begin{bmatrix} g_{11}(t) & g_{12}(t) & \cdots & g_{1N}(t) \\ & g_{22}(t) & \vdots & g_{2N}(t) \\ & & \vdots & \vdots \\ & & & g_{NN}(t) \end{bmatrix} \end{aligned} \tag{4.25}$$

in one computational step. The LS_y model-set $\Theta(t,N)$ pertains to the following assumption:

Assumption 4.1 *The plant order is assumed to be higher than the model-set $\Theta(t,N)$, or in other words, $\Theta(t,N)$ is assumed to be underparameterized and therefore the plant does not belong to the model-set $\Theta(t,N)$.*

Assumption-4.1 is true in most real situations (such as in chemical processes), where the plant dynamics are complex and generally of very high order. Therefore estimation of the orthonormal series model is concerned with the identification of a suitable finite and reduced-order model that adequately approximates the plant dynamics. In view of ***Assumption***-4.1, it should be recognized that the model-set $\Theta(t,N)$ is non-unique as it depends on several identification design conditions such as: ***a priori*** system information, system noise and design factors such as the choice of model order, nature of excitation, data length and data pre-filtering. Let these identification design conditions be symbolically denoted by $\mathcal{F}$.

It should be noted that the structure of the data vector defined by equation (4.23) is different from the AUDI-ARMAX formulation [16, 17]. Specifically the signal $-y(t)$ is placed at the beginning of the data vector rather than at the end. This enables equation (4.23) to be recursively expressed as:

$$\phi(t,N) = \left[\phi^T(t,N-1), \psi_N(t)\right]^T \tag{4.26}$$

Rearrangement of equation (4.24) leads to:

$$\begin{aligned}\psi_N(t) &= \frac{-1}{g_{NN}(t)}\left[1,\theta^T(t,N-1)\right]\phi(t,N-1) \\ &= -[\xi_1(t,N),\xi_2(t,N),\ldots,\xi_N(t,N)]\phi(t,N-1) \\ &= -\xi^T(t,N)\phi(t,N-1) \end{aligned} \tag{4.27}$$

where the new set of parameters $\xi(t,N)$ is:

$$\xi(t,N) = \frac{1}{g_{NN}(t)}\left[1,\theta^T(t,N-1)\right]^T \tag{4.28}$$

Analogous to $\Theta(t,N)$ in (4.25), the set of AUDI parameters $[\xi(t,1),\xi(t,2),\ldots,\xi(t,N)]$ is compactly denoted by $\Xi(t,N)$ in this paper.

4.4.2 LS properties of the AUDI estimated model parameters $\xi(t, N)$

For a specified identification design condition $\mathcal{F}$, let the LS_y parameter set $\Theta(t, N)$ represent the *reference* model-set. Consequently, the parameter set $\Xi(t, N)$ which is related to $\Theta(t, N)$ via equation (4.28) for model orders 1 to N also corresponds to the AUDI *reference* parameter set. Let the LS solution for the AUDI parameter vector $\xi(t, N)$ be denoted by $\hat{\xi}(t, N)$, then from equation (4.27), the estimated last basis signal corresponding to $\hat{\xi}(t, N)$ is expressed as:

$$\hat{\psi}_N(i) = -\phi^T(i, N-1)\hat{\xi}(t, N) \tag{4.29}$$

From equations (4.27) and (4.29) it is clear that the LS solution $\hat{\xi}(t, N)$ is obtained by minimizing the following objective function:

$$J(t, \xi, N) = \sum_{i=1}^{t} \left[\psi_N(i) - \hat{\psi}_N(i)\right]^2 \tag{4.30}$$

Since the above objective function is different from the classical LS identification (*i.e.* LS_y), the LS estimates $\hat{\xi}(t, N)$ of $\xi(t, N)$ are therefore stated in the following as:

Lemma 4.1 *The LS estimate $\hat{\xi}(t, N)$ is given by:*

$$\hat{\xi}(t, N) = -\left[\sum_{i=1}^{t} \phi(i, N-1)\phi^T(i, N-1)\right]^{-1} \sum_{i=1}^{t} \phi(i, N-1)\psi_N(i) \tag{4.31}$$

Proof: The LS estimates $\hat{\xi}(t, N)$ of $\xi(t, N)$ are obtained by substituting equation (4.29) in (4.30) and minimizing the resulting equation with respect to $\xi(t, N)$ as shown below:

$$\left.\frac{\partial J(t, \xi, N)}{\partial \xi(t, N)}\right|_{\hat{\xi}(t,N)} = \frac{\partial}{\partial \xi(t, N)} \left.\left[\sum_{i=1}^{t} \left(\psi_N(i) + \phi^T(i, N-1)\xi(t, N)\right)^2\right]\right|_{\hat{\xi}(t,N)} = 0 \tag{4.32}$$

Equation (4.32) results in:

$$\sum_{i=1}^{t} \phi(i, N-1)\phi^T(i, N-1)\hat{\xi}(t, N) + \sum_{i=1}^{t} \phi(i, N-1)\psi_N(i) = 0 \tag{4.33}$$

The best LS estimate $\hat{\xi}(t, N)$ of $\xi(t, N)$ is therefore obtained as shown in equation (4.31).
□

Corollary 4.1 *The minimum value of $J(t, \xi, N)$ in equation (4.30) is given by:*

$$J(t, N) = \sum_{i=1}^{t} \psi_N^2(i) - \hat{\xi}^T(t, N) \left[\sum_{i=1}^{t} \phi(i, N-1)\phi^T(i, N-1)\right] \hat{\xi}(t, N) \tag{4.34}$$

4.4.3 AUDI formulation for the orthonormal series models

The data vector $\phi(t, N)$ in equation (4.26) is used to form the following *Data Product Moment Matrix* (DPMM), $S(t, N)$, to obtain multiple orthonormal model estimation as follows:

$$\begin{aligned} S(t,N) &= \sum_{i=1}^{t} \phi(i,N)\phi^T(i,N) \\ &= \begin{bmatrix} \sum_{i=1}^{t} \phi(i,N-1)\phi^T(i,N-1) & \sum_{i=1}^{t} \phi(i,N-1)\psi_N(i) \\ \sum_{i=1}^{t} \psi_N(i)\phi^T(i,N-1) & \sum_{i=1}^{t} \psi_N^2(i) \end{bmatrix} \end{aligned} \tag{4.35}$$

The inverse of DPMM *i.e.* $S^{-1}(t, N)$ is called the ***Information Accumulation Matrix*** (IAM) [16, 17]. The LDL$^{\mathrm{T}}$ decomposition of $S(t, N)$ in equation (4.35) yields [16, 17]:

$$S(t,N) = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} = \begin{bmatrix} I_1 & 0 \\ B^TA^{-1} & I_2 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} I_1 & 0 \\ B^TA^{-1} & I_2 \end{bmatrix}^T \tag{4.36}$$

From equations (4.35) and (4.36) it follows that:

$$A = \sum_{i=1}^{t} \phi(i,N-1)\phi^T(i,N-1) = S(t,N-1) \tag{4.37}$$

$$B = \sum_{i=1}^{t} \phi(i,N-1)\psi_N(i) \tag{4.38}$$

$$D = \sum_{i=1}^{t} \psi_N^2(i) \tag{4.39}$$

and

$$\Delta = D - B^TA^{-1}B \tag{4.40}$$

From equations (4.27), (4.31), (4.37) and (4.38), B^TA^{-1} can be expressed as:

$$\begin{aligned} B^TA^{-1} &= \sum_{i=1}^{t} \psi_N(i)\phi^T(i,N-1) \left[\sum_{i=1}^{t} \phi(i,N-1)\phi^T(i,N-1)\right]^{-1} \\ &= \sum_{i=1}^{t} \psi_N(i)\phi^T(i,N-1) \left[S^{-1}(t,N-1)\right] = -\tilde{\xi}^T(t,N) \end{aligned} \tag{4.41}$$

Furthermore from equations (4.29), (4.30) and (4.34), the term Δ as defined by equation (4.40) can be shown to be the minimum value of the loss function $J(t, N)$ as follows:

$$\begin{aligned}\Delta &= \sum_{i=1}^{t}\psi_N^2(i) + \hat{\xi}^T(t,N)\sum_{i=1}^{t}\phi(i,N-1)\psi_N(i)\\ &= \sum_{i=1}^{t}\psi_N^2(i) - \hat{\xi}^T(t,N)\left[\sum_{i=1}^{t}\phi(i,N-1)\phi^T(i,N-1)\right]\hat{\xi}(t,N) = J(t,N)\end{aligned} \tag{4.42}$$

Equation (4.36) can be used for the nested LDL$^{\mathrm{T}}$ decomposition of $S(t, N-1)$ in equations (4.35) or (4.37). The following expression is obtained after successive LDL$^{\mathrm{T}}$ decompositions of $S(t, N-1)$:

$$S(t,N) = \mathcal{L}(t,N)\mathcal{D}(t,N)\mathcal{L}^T(t,N) \tag{4.43}$$

where $$\mathcal{L}(t,N) = L_N L_{N-1} \ldots L_2 L_1 \tag{4.44}$$

The argument t is omitted from the right hand side matrices $L_k, k = 1 \ldots N$ in the above equation (4.44) for brevity and convenience. The right hand side matrix L_k in equation (4.44) is then given by:

$$L_k = \begin{bmatrix} I_k & 0_{k\times 1} & 0_{k\times(N-k)} \\ -\hat{\xi}^T(t,k) & I_1 & 0_{1\times(N-k)} \\ 0_{(N-k)\times k} & 0_{(N-k)\times 1} & I_{N-k} \end{bmatrix} \tag{4.45}$$

where I_m and $0_{m\times n}$ in the above equation (4.45) are the identity matrix of dimension m and matrix of zeros of size $m \times n$ respectively. From equations (4.36), (4.40), (4.42) and (4.43) the diagonal matrix $\mathcal{D}(t, N)$ can be shown to be:

$$\mathcal{D}(t,N) = \begin{bmatrix} J(t,0) & & & & & \\ & J(t,1) & & & & \\ & & J(t,2) & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & J(t,N) \end{bmatrix} \tag{4.46}$$

Similarly the UDU$^{\mathrm{T}}$ decomposition of IAM can be performed as shown below [16, 17]:

$$\begin{aligned}[S(t,N)]^{-1} &= \left[\mathcal{L}(t,N)\mathcal{D}(t,N)\mathcal{L}^T(t,N)\right]^{-1}\\ &= \left[L_N L_{N-1}\cdots L_2 L_1\ [\mathcal{D}(t,N)]\ L_1^T L_2^T \cdots L_{N-1}^T L_N^T\right]^{-1}\end{aligned}$$

$$
\begin{aligned}
&= \left[L_N^{-T} L_{N-1}^{-T} \cdots L_2^{-T} L_1^{-T}\right] \ [\mathcal{D}(t,N)]^{-1} \ \left[L_1^{-1} L_2^{-1} \cdots L_{N-1}^{-1} L_N^{-1}\right] \\
&= [U_N U_{N-1} \cdots U_2 U_1] \ [\mathcal{D}(t,N)]^{-1} \ \left[U_1^T U_2^T \cdots U_{N-1}^T U_N^T\right] \\
&= \mathcal{U}(t,N)[\mathcal{D}(t,N)]^{-1} \mathcal{U}^T(t,N) \qquad (4.47)
\end{aligned}
$$

The elements in $\mathcal{L}(t,N)$ do not directly correspond to the AUDI estimated parameter set $\hat{\Xi}(t,N) = [\hat{\xi}(t,1), \hat{\xi}(t,2), \ldots, \hat{\xi}(t,N)]$. A transformation $\mathcal{U}(t,N) = [\mathcal{L}(t,N)]^{-T}$ is required as shown in equation (4.47) in order to obtain the AUDI parameter set $\hat{\Xi}(t,N)$. This is also illustrated by an example in *appendix-1*.

The LS estimated AUDI parameter set $\hat{\Xi}(t,N)$ as such cannot be used directly for modelling. Therefore it is required to convert $\hat{\Xi}(t,N)$ into a usable model-set having the form $\Theta(t,N)$. Let such a converted model-set which is derived from $\hat{\Xi}(t,N)$, be denoted by $\hat{\Theta}(t,N)$. It can then be shown that $\hat{\Theta}(t,N)$ is obtained using simple algebraic manipulations of $\hat{\Xi}(t,N)$, because the AUDI decomposition of $S(t,N)$ corresponds to solving:

$$
\begin{bmatrix}
1 & & & & \\
\hat{\xi}_1(t,1) & 1 & & & \\
\hat{\xi}_1(t,2) & \hat{\xi}_2(t,2) & 1 & & \\
\vdots & \vdots & \vdots & \ddots & \\
\hat{\xi}_1(t,N) & \cdots & \cdots & \hat{\xi}_N(t,N) & 1
\end{bmatrix}
\begin{bmatrix}
-y(t) \\ \psi_1(t) \\ \psi_2(t) \\ \vdots \\ \psi_N(t)
\end{bmatrix}
= 0 \qquad (4.48)
$$

Further, the conversion from $\hat{\Xi}(t,N)$ to $\hat{\Theta}(t,N)$ is illustrated by considering the k^{th} row of the above equation (4.48):

$$
-\hat{\xi}_1(t,k)y(t) + \hat{\xi}_2(t,k)\psi_1(t) + \cdots + \hat{\xi}_k(t,k)\psi_{k-1}(t) + \psi_k(t) = 0 \qquad (4.49)
$$

which can then be rearranged to:

$$
y(t) = \left(\frac{1}{\hat{\xi}_1(t,k)}\right)\left(\hat{\xi}_2(t,k)\psi_1(t) + \cdots + \hat{\xi}_k(t,k)\psi_{k-1}(t) + \psi_k(t)\right) \qquad (4.50)
$$

Therefore comparing equation (4.50) with a combination of equations (4.24), (4.27), (4.28) and (4.48), it leads to:

$$
\hat{\theta}(t,k) = \left[\frac{\hat{\xi}_2(t,k)}{\hat{\xi}_1(t,k)}, \frac{\hat{\xi}_3(t,k)}{\hat{\xi}_1(t,k)}, \ldots, \frac{\hat{\xi}_k(t,k)}{\hat{\xi}_1(t,k)}, \frac{1}{\hat{\xi}_1(t,k)}\right]^T = [\hat{g}_{1k}(t), \hat{g}_{2k}(t), \ldots, \hat{g}_{kk}(t)]^T \qquad (4.51)
$$

In this way the parameters of all models of order $k = 1, \ldots, N$ can be determined. Niu *et al.* have recently shown that $\Theta(t,N)$ can be obtained by performing LU decomposition of a non-symmetric DPMM which is obtained as: $S^*(t,N) = \sum_{i=1}^{t} \phi(i,N)[h^T(i,N), -y(i)]^T$ [20]. The approach proposed here results in $\hat{\Theta}(t,N)$ by applying the AUDI method to $S(t,N)$. $S(t,N)$ and $S^*(t,N)$ are related to each other, *i.e.* the first column of $S(t,N)$

corresponds to the last column of $S^*(t, N)$. The $S(t, N)$ matrix is symmetric and can be decomposed by using UDU$^{\mathrm{T}}$ or LDL$^{\mathrm{T}}$ decompositions. $S^*(t, N)$ on the other hand is non-symmetric and can only be decomposed using LU decomposition. The AUDI method presented in this paper complements the LS_y solution by confirming an appropriate order for the model *i.e.* as the model order approaches the plant, $\hat{\theta}(t, N) \rightarrow \theta(t, N)$.

4.4.4 Effect of noise and modelling error in the AUDI method

This subsection analyses the effect of noise and modelling error in the performance of the AUDI estimation method. It is shown in this section that noise and modelling errors have detrimental effect in the AUDI estimation procedure. The way to circumvent some of these problems are also discussed here.

In reality it is difficult to meet Parseval's condition given by definition-4.17, as the plant order is usually considered to be very large *i.e.* $N \rightarrow \infty$ in comparison to the model order. Therefore for most practical purposes and also according to ***assumption***-4.1, the LHS of the above equation (4.11) exceeds the RHS term *i.e.* it obeys Bessel's inequality as given by definition-4.18 [5]. It is obvious that the systems belonging to the class of Bessels inequality are underparameterized and consequently they introduce a modelling or bias error. Let this error be denoted by the term $b(t)$. In addition, the process measurements are usually corrupted by the noise sequence $e(t)$ satisfying the assumption:

Assumption 4.2 *The noise, $e(t)$, is zero mean and is a sequence of normally distributed random white measurement noise.*

According to ***assumption***-4.1, the plant does not belong to model-set because the models are underparameterized or in other words they obey ***Bessel's inequality***. Therefore the analysis of the proposed AUDI method considers the effect of modelling error $b(t)$ and noise $e(t)$ (*i.e.* ***assumption***-4.2) as shown in the following:

$$\begin{aligned} y(t) &= \sum_{k=1}^{N} g_{kN}(t)\psi_k(t) + b(t) + e(t) \\ &= h^T(t, N)\theta(t, N) + b(t) + e(t) \end{aligned} \tag{4.52}$$

For the AUDI formulation, a combination of equations (4.27) and (4.52) yields:

$$-\psi_N(t) = \phi^T(t, N-1)\xi(t, N) + \xi_1(t, N)e(t) + \xi_1(t, N)b(t) \tag{4.53}$$

For the above equation (4.53), the LS estimate of $\xi(t, N)$ for the AUDI solution is given by the following *lemma*:

Lemma 4.2 *The least square estimate for $\xi(t, N)$ corresponding to equation (4.53) (i.e. the AUDI method) is given by:*

$$\begin{aligned}\mathcal{E}[\hat{\xi}(t,N)] &= \xi(t,N) + \mathcal{E}\left\{\left[\sum_{i=1}^{t}\phi(i,N-1)\phi^T(i,N-1)\right]^{-1}\right. \\ &\qquad \left.\left[\sum_{i=1}^{t}(e(i)+b(i))\xi_1(t,N)\phi^T(i,N-1)\right]\right\} \\ &= \xi(t,N) + \mathcal{E}\left\{S^{-1}(t,N-1)\left[\sum_{i=1}^{t}(e(i)+b(i))\xi_1(t,N)\phi^T(i,N-1)\right]\right\}\end{aligned} \tag{4.54}$$

where $\mathcal{E}$ is the expectation operator.

Proof: Substituting equation (4.53) in (4.31) and taking expectation leads to the proof. □

The following *corollaries* and *remarks* then apply to *lemma*-4.2:

Remark 4.1 *From equation (4.4.4) it is clear that for a "complete" model (i.e. $b(t) = 0$) and a noise-free process (i.e. $e(t) = 0$), $\hat{\xi}(t,N) = \xi(t,N)$, i.e. the estimates $\hat{\xi}(t,N)$ converge to their 'true' values $\xi(t,N)$ i.e. unbiased estimates are obtained. For such a case the AUDI and LS_y solutions are equivalent as illustrated by an example in appendix-2.*

Corollary 4.2 *The AUDI and LS_y solutions are not equivalent for the (underparameterized) models that obey Bessel's inequality.*

Proof: For the models that obey Bessel's inequality, $b(t) \neq 0$. Therefore from equations (4.31), and (4.4.4) parameters $\hat{\xi}(t,N)$ are biased by $\mathcal{E}\{S^{-1}(t,N-1)\sum_{i=1}^{t}b(i)\xi_1(t,N)\phi^T(i,N-1)\}$.

Remark 4.2 *Corollary-4.2 is true for even noise-free processes i.e. $e(t) = 0$. With the increase in model order the AUDI solution approaches the LS_y solution, because $b(t) \rightarrow 0$.*

Corollary 4.3 *In presence of noise $e(t)$, the AUDI solution is biased even for a "complete" model.*

Proof: From equation (4.4.4) it is clear that the error correlation term: $\mathcal{E}\{\sum_{i=1}^{t}e(i)\xi_1(t,N)\phi^T(i,N-1)\} \neq 0$ (even for $t \rightarrow \infty$) because $\phi(i,N-1)$ is correlated with $e(i)$ due to the presence of $y(i)$ in $\phi(i,N-1)$.

Remark 4.3 *For any model order k, where $1 \leq k \leq N$, it is required that $\hat{g}_{kk}(t) > 0$ for the AUDI method to work. This condition also holds good for the LU decomposition of $S^{\star}(t,N)$ [20]. It is unlikely that a situation such as $\hat{g}_{kk}(t) = 0$ will arise in practice because the* *plant order is high compared to the model order according to assumption-4.1 and also the knowledge of exact location of the plant dominant poles required by the orthonormal models to generate the basis signals is exactly not known.*

Remark 4.4 *Small values of $\hat{g}_{kk}(t)$, $1 \le k \le N$, also tend to amplify $b(t)$ and $e(t)$ in equation (4.4.4) because $\hat{\xi}_1(t,k)$ is inverse of $\hat{g}_{kk}(t)$. This is perhaps a major limitation of the proposed method; however, the approach using LU decomposition of $S^*(t,N)$ does not have such a constraint. Therefore for noisy data, $y(t)$ in $\phi(t,N)$ (in equation (4.23)) is replaced by $\hat{y}(t,N')$ which is obtained via the LS_y method for some large N'. In many applications as in process control, use of such higher order models (i.e. of order N') is not desirable due to practical reasons, but they can be used as a reference by the AUDI method to obtain a lower order model-set $\hat{\Theta}(t,N)$. It is shown in the succeeding sections, that such an approximation method can work well for orthonormal models. This procedure complements the LS_y solution by confirming an appropriate order for the model.*

Remark 4.5 *UDU^{T} decomposition of the IAM matrix (i.e. $S^{-1}(t,N)$) is computationally the most efficient way to determine the $\mathcal{U}(t,N)$ matrix. But this method cannot be used when $S(t,N)$ is rank deficient. LU decomposition of $S(t,N)$ is another computationally efficient but an indirect way to obtain the $\mathcal{U}(t,N)$ matrix. LU decomposition of the $S(t,N)$ results in the $\mathcal{L}(t,N)$ matrix which can then be used to obtain the AUDI parameter set by using the transformation: $\mathcal{U}(t,N) = \mathcal{L}^{-T}(t,N)$. LU decomposition can be used for rank deficient $S(t,N)$ as this algorithm does not involve any matrix inversions [26]. The LDL^{T} decomposition of $S(t,N)$ as shown in Section 4.4 is not computationally efficient as it involves N matrix inversions, where N is the maximum order of the model. The LDL^{T} decomposition shown in equations (4.36) through (4.47) is for analytical derivation purposes only.*

Remark 4.6 *From remark-4.4 it can be seen that if $\hat{y}(t,N')$ is used in place of $y(t)$ to compute $S(t,N)$, then $S(t,N)$ becomes rank deficient (by one column) if $N' \le N$. It is recommended that the LU decomposition be used for such a rank deficient $S(t,N)$. However if UDU^{T} decomposition is to be used for rank deficient $S(t,N)$, then $\hat{y}(t,N')$ should be replaced by $\hat{y}(t,N') + \lambda(\hat{y}(t,N') - y(t))$ in the regressor $\phi(t,N)$. The choice of λ is crucial because λ has to be small enough to suppress the effect of noise in $\phi(t,N)$ and yet large enough to remove the effect of collinearity in $S(t,N)$. For example, a λ value of 10^{-5} was adequate to handle the examples illustrated in the ensuing sections.*

4.5 Simultaneous estimation of Laguerre models with real poles

The Laguerre function model with a real pole $a, (|\, a \,| < 1)$ is used to approximate the dynamics of an overdamped process. The real pole a in the Laguerre series represents the dominant time constant of the process. The discrete Laguerre functions for a real pole in the z-domain are defined as [9, 6]:

$$\Psi_k(z) = L_k(z) = \frac{\sqrt{1-a^2}}{z-a}\left(\frac{1-az}{z-a}\right)^{k-1} \tag{4.55}$$

The above Laguerre filter pertains to discrete models where a unit delay is implicit as discussed in the previous section. Therefore any discrete linear stable system with the transfer function $G(z)$ can be expressed as a weighted sum of the Laguerre filter series as:

$$G(z) = \sum_{k=1}^{\infty} g_k \; L_k(z) \tag{4.56}$$

where the coefficients $g_k, (k = 1, 2, 3, \ldots)$ are called the Laguerre gains. Using equations (4.9) and (4.19), the estimated output for the N^{th} order Laguerre model can be written as:

$$\hat{y}(t, N) = \sum_{k=1}^{N} g_{kN}(t) \; l_k(t) \tag{4.57}$$

where $l_k(t) = L_k(q)u(t)$. The AUDI solution for Laguerre models is obtained by substituting $\psi_k(t) = l_k(t)$ and forming the data vector shown in equation (4.23). The AUDI method described in the earlier section (*i.e.* equations (4.23) through (4.51) and (4.4.4)) is then used to obtain the parameter set $\hat{\Theta}(t, N)$ via the $\mathcal{U}(t, N)$ matrix for the Laguerre model. A comparison of the estimated Laguerre gains between the LS_y and AUDI methods is shown in the following section via a simulation example.

4.5.1 Simulation Example

Consider a third order noise-free overdamped stable plant: $G(s) = 1/(s+1)(3s+1)(5s+1)$, whose discrete equivalent for a sample time of $Ts = 1$ is:

$$G(z) = \frac{0.0077z^{-1} + 0.0212z^{-2} + 0.0036z^{-3}}{1 - 1.9031z^{-1} + 1.1514z^{-2} - 0.2158z^{-3}} \tag{4.58}$$

The batch LS_y solution for a square wave type of excitation for 1 to 4^{th} order Laguerre network models and their corresponding loss functions $\mathcal{J}_y(t)$ are shown in Table 4.10 for $a = 0.8434$. The AUDI solution for this problem is shown in Table 4.10 for comparison. The loss functions in Table 4.10 indicate that a 3^{rd} order Laguerre filter model adequately represents the dynamics of the system.

The UDU$^{\mathrm{T}}$ decomposition of the IAM results in the following $\hat{\xi}(t, N)$ parameter matrix $\mathcal{U}(t, N)$, and the loss function matrix $\mathcal{D}(t, N)$:

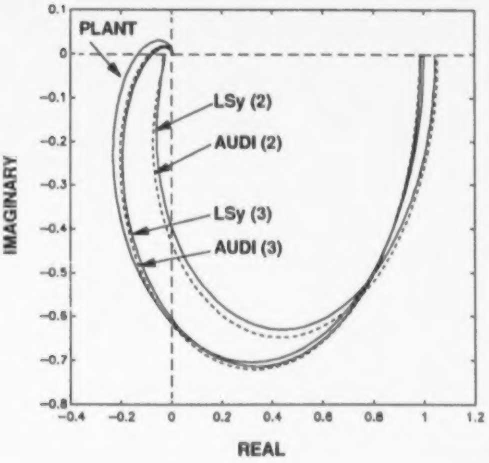


Figure 4.1: Comparison between AUDI and LS_y methods for 2^{nd} and 3^{rd} order Laguerre models.

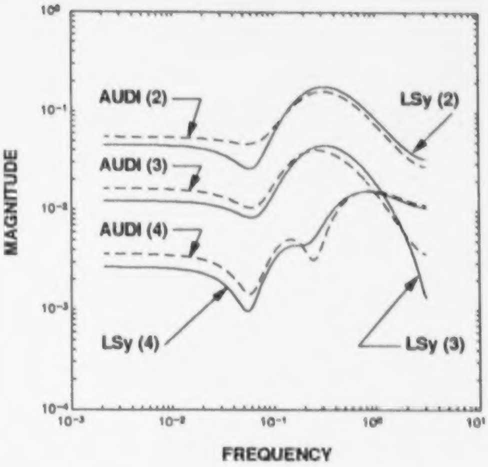


Figure 4.2: Magnitude Spectrum of the modelling errors for the AUDI and LS_y based Laguerre models for orders 2 to 4.

$$\mathcal{U}(t,4) = \begin{pmatrix} 1 & 3.3988 & 9.1601 & -21.7299 & -84.5853 \\ 0 & 1 & 1.8311 & -3.8239 & 15.1486 \\ 0 & 0 & 1 & -3.3769 & 13.4076 \\ 0 & 0 & 0 & 1 & -4.7788 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.59}$$

$$\mathcal{D}(t,4) = \mathsf{diag}\,(150.9492, \quad 100.2522, \quad 68.7863, \quad 23.7327, \quad 19.8) \tag{4.60}$$

Equation (4.51) is used to calculate the model-set $\hat{\Theta}(t,N)$ from equation (4.59) as shown in Table 4.10. As an example the parameters calculated for the 3^{rd} order Laguerre model by the AUDI method are:

$$\begin{aligned} \hat{\theta}(t,3) &= \left[\frac{\hat{\xi}_2(t,3)}{\hat{\xi}_1(t,3)}, \frac{\hat{\xi}_3(t,3)}{\hat{\xi}_1(t,3)}, \frac{1}{\hat{\xi}_1(t,3)}\right]^T \\ &= \left[\frac{-3.8239}{-21.7299}, \frac{-3.3769}{-21.7299}, \frac{1}{-21.7299}\right]^T \\ &= [0.1760, \quad 0.1554, \quad -0.0460]^T \\ &= [\hat{g}_{13}(t), \quad \hat{g}_{23}(t), \quad \hat{g}_{33}(t)]^T \end{aligned} \tag{4.61}$$

Similar to the LS_y results, the AUDI results (Table 4.10 and equation (4.60)) also indicate that a 3^{rd} order Laguerre filter model adequately represents the process dynamics. From Tables 4.10 and 4.10 it is observed that with the increase in the order of Laguerre

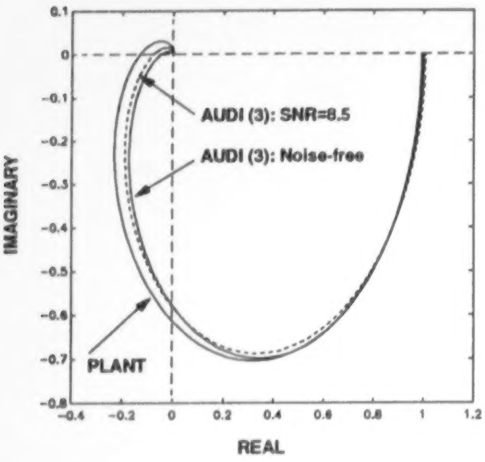


Figure 4.3: Effect of noise in the estimation of Laguerre model via the AUDI method

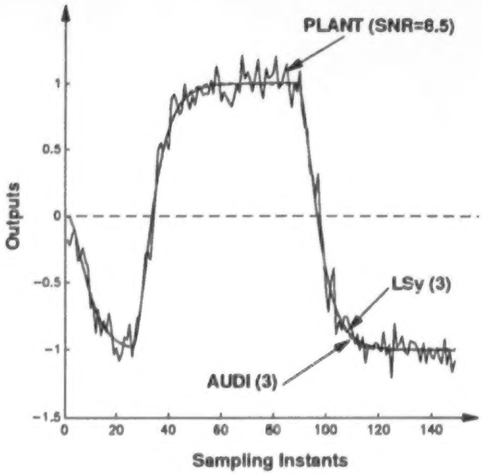


Figure 4.4: Comparison between AUDI and LS_y methods for 3^{rd} order Laguerre models for noisy data.

model the difference in Laguerre gains or loss function as obtained from AUDI and LS_y methods diminishes considerably.

Figure 4.1 compares the Nyquist plots between the true plant and the estimated models obtained from the 2^{nd} and 3^{rd} order Laguerre filter models obtained via the AUDI and LS_y methods. The figure shows little difference in the Nyquist plots of LS_y and AUDI based models for both 2^{nd} and 3^{rd} order Laguerre filter models.

The 3^{rd} or 4^{th} columns of Tables 4.10 and 4.10 consistently indicate that the time domain performances of the LS_y models are better than the AUDI based models. However from the Nyquist plots in Figure 4.1 it is difficult to conclude if the LS_y models are better than the AUDI based models. The euclidean norm of the modelling errors shown by the last column in Tables 4.10 and 4.10 infact indicate that the AUDI based 2^{nd} and 3^{rd} order Laguerre models are better than the LS_y based models in the frequency domain. The magnitude spectrum of the modelling errors for the AUDI and LS_y based models are shown in Figure 4.2, which also show for this example that: (a) $||G(\omega) - \hat{G}(\omega)||_\infty$ is lower for the AUDI based method; (b) at the lower frequencies LS_y based models are better; and (c) AUDI models are generally better at the higher frequencies. Although these results cannot be generalized, nevertheless one may conclude that: if the AUDI and LS_y models are close, then it is difficult to decide which of these two models is better in the frequency domain.

In Figure 4.3, the Nyquist plots of 3^{rd} order Laguerre filter models for noise-free and noisy (SNR=8.5) data are compared with the models obtained by the AUDI method. The result shows that the AUDI method with appropriate modifications discussed under *remark*-4.4 performs well even in the presence of significant noise. This confirms as shown in Figure 4.4, that the AUDI method with the above modification can be used even in the presence of noisy process data.

4.6 Simultaneous estimation of Laguerre models with complex poles

The Laguerre series with a complex pole is a special case of the more general discrete Kautz series described in the next section. Fu and Dumont [6] have shown the following formulation for the discrete Laguerre functions for a complex pole $a, (\mid a \mid < 1)$, to approximate the dynamics of underdamped systems:

$$\begin{aligned} \Psi_{2k-1}(z) &= \Psi_1(z)\Psi^{k-1}(z) \\ \Psi_{2k}(z) &= \Psi_2(z)\Psi^{k-1}(z) \end{aligned} \tag{4.62}$$

$$\begin{aligned} \text{where: } \Psi_1(z) &= N_1\sqrt{1-r^2}\left(\tfrac{z-m_1}{z^2-2\alpha z+r^2}\right),\ \Psi_2(z) = N_2\sqrt{1-r^2}\left(\tfrac{1-m_2 z}{z^2-2\alpha z+r^2}\right) \\ \Psi(z) &= \left(\tfrac{1-2\alpha z+r^2z^2}{z^2-2\alpha z+r^2}\right);\ r=\sqrt{\alpha^2+\beta^2};\ m_1=\alpha;\ m_2=\alpha\left(\tfrac{1-\alpha^2-\beta^2}{1-\alpha^2+\beta^2}\right) \\ N_1^2 &= \tfrac{(1+\alpha^2+\beta^2)^2-4\alpha^2}{(1+\alpha^2+\beta^2)(1+m_1^2)-4m_1\alpha};\ N_2^2 = \tfrac{(1+\alpha^2+\beta^2)^2-4\alpha^2}{(1+\alpha^2+\beta^2)(1+m_2^2)-4m_2\alpha} \end{aligned} \tag{4.63}$$

and where α and β are real and imaginary components of the dominant complex pole a. As described in Section 4.4, the regressor $\phi(t, N)$ can be formed to solve the AUDI problem. Since the Laguerre model for complex pole is a special realization of the Kautz model, simulation results for this section are lumped with the example presented for the Kautz model.

4.7 Simultaneous estimation of Kautz models

As stated in the previous section, Kautz models are used to describe resonant systems. Kautz models can be represented as [9, 11]:

$$\begin{aligned} \Psi_{2k-1}(z) &= \frac{\sqrt{1-c^2}(z-b)}{z^2+b(c-1)z-c}\left(\frac{-cz^2+b(c-1)z+1}{z^2+b(c-1)z-c}\right)^{k-1} \\ \Psi_{2k}(z) &= \frac{\sqrt{(1-c^2)(1-b^2)}}{z^2+b(c-1)z-c}\left(\frac{-cz^2+b(c-1)z+1}{z^2+b(c-1)z-c}\right)^{k-1} \end{aligned} \tag{4.64}$$

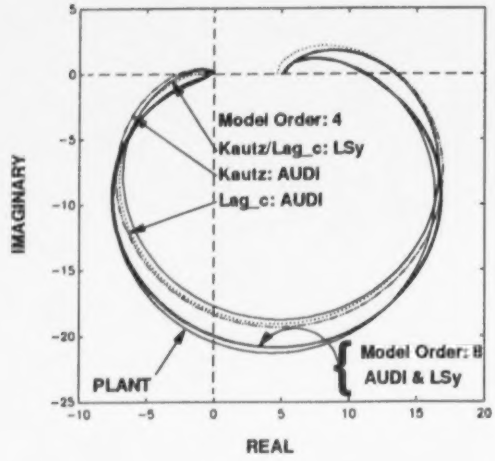


Figure 4.5: Comparison between AUDI and LS_y methods for 4^{th} and 8^{th} order Kautz and Lag_c models.

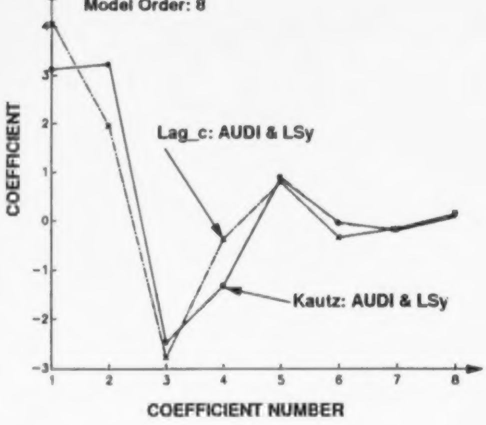


Figure 4.6: Parameter profile for 8^{th} order Kautz and Lag_c models obtained via the AUDI and LS_y methods.

where b and c ($|b| < 1, |c| < 1$) are obtained by solving $z^2 + b(c-1)z^2 - c = 0$ such that z corresponds to the dominant (complex) pole of the system. The AUDI estimation of Kautz and Laguerre models with complex poles is considered in the following illustrative example.

4.7.1 Simulation Example

Consider an underdamped noise-free stable plant having zeros at $[0.1 \pm j0.2, 0.2 \pm j0.1]$ and poles at $[0.8 \pm j0.2, 0.88 \pm j0.3]$ which corresponds to a peak magnitude of 22.26 at a normalized frequency $\omega/\omega_n = 0.1033$. Figure 4.5 compares the Nyquist plots for 4^{th} and 8^{th} order Kautz and Laguerre models with complex poles (denoted by *Lag_c*) obtained via the AUDI and LS_y methods. A small difference is observed between the Kautz-AUDI and Lag_c-AUDI models for the 4^{th} order models. However for the 8^{th} order model, the difference between LS_y and AUDI models is not apparent.

Figure 4.6 shows parameter profile for the 8^{th} order Kautz and Lag_c models as obtained by the AUDI and LS_y approaches. The figure shows that the AUDI and LS_y parameters are in good agreement. Kautz and Lag_c models essentially give the same model but their parameter profiles are different because of the difference between their basis signals. Figure 4.7 shows that 8^{th} order Kautz and Lag_c models obtained via the LS_y and AUDI methods are in good agreement for noisy data having a signal-to-noise ratio of 8.5.

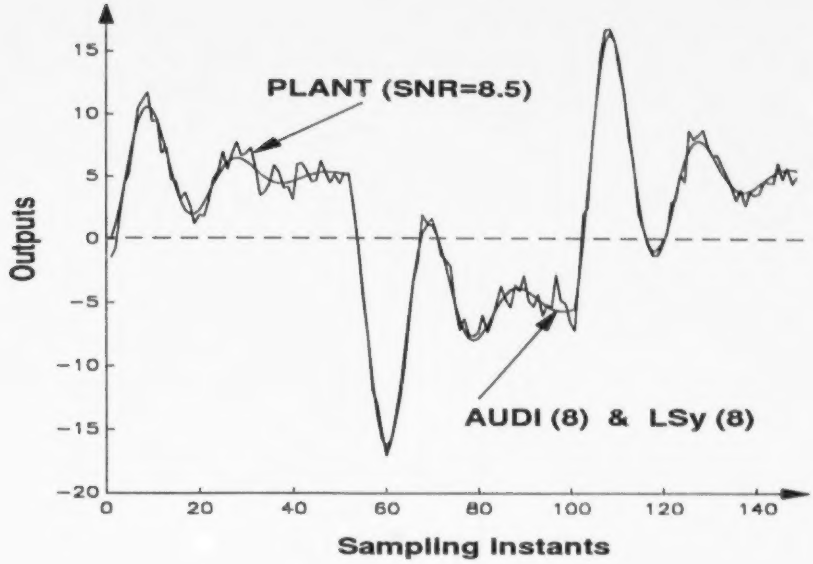


Figure 4.7: Comparison between AUDI and LS_y methods for 8^{th} order Kautz/Lag_c models for noisy data.

4.8 Simultaneous estimation of FIR Models

The FIR model is a special case of orthonormal functions described in the preceding sections [9]. The basis function $\Psi_k(z)$ for the FIR model is defined as:

$$\Psi_k(z) = z^{-k} \tag{4.65}$$

Several industrial predictive controllers used in the chemical industries are based on the FIR/step-response models. Generally a large number of coefficients are required to describe the dynamics of a process using FIR or step-response models which is typically of the order of 20 to 60 or even more, depending on the sampling time and nature of the process dynamics. However this allows the FIR models to capture unusual process dynamics. The method of AUDI is used here to simultaneously estimate the coefficients of all M to N^{th} order FIR or step-response model, where M and N can be fairly large. A linear stable transfer function $G(z)$ can be expressed in the infinite impulse response form by:

$$G(z) = \sum_{k=1}^{\infty} f_k \, z^{-k} \tag{4.66}$$

or in the step-response form as:

$$G(z) = \left(1 - z^{-1}\right) \sum_{k=1}^{\infty} s_k \, z^{-k} \tag{4.67}$$

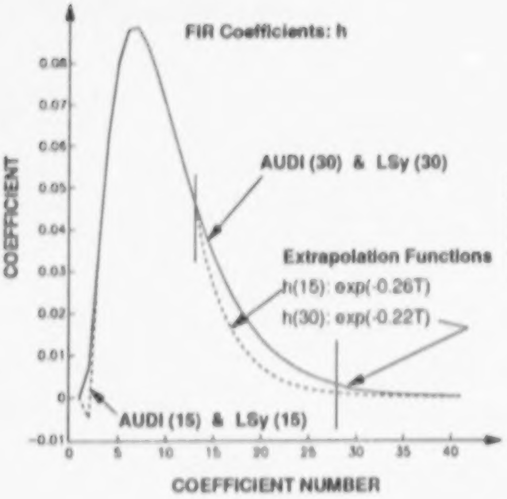


Figure 4.8: A comparison between 15th and 30th order FIR coefficients obtained using AUDI and LS_y methods.

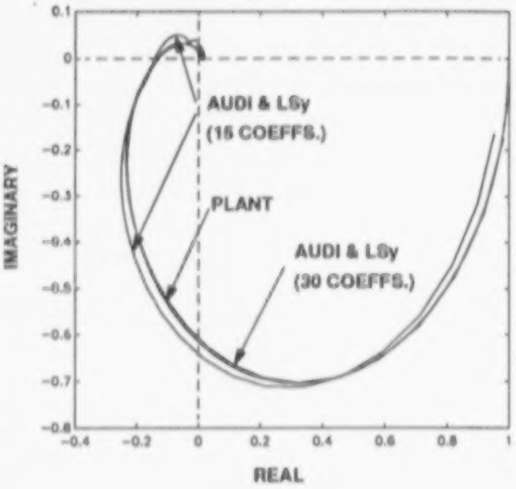


Figure 4.9: A comparison between 15th and 30th order FIR models obtained using AUDI and LS_y methods.

where f_k and s_k are the FIR and step-response coefficients respectively. The basis signal $\psi_k(t)$ for the FIR and step-response models are $u(t-k)$ and $\Delta u(t-k)$ respectively. Therefore the data vector for the N^{th} order FIR-AUDI formulation becomes:

$$\phi(t,N) = [-y(t), u(t-1), u(t-2), \dots, u(t-N)]^T \tag{4.68}$$

similarly the data vector for the AUDI formulation of the step-response model is given by:

$$\phi(t,N) = [-y(t), \Delta u(t-1), \Delta u(t-2), \dots, \Delta u(t-N)]^T \tag{4.69}$$

The AUDI method described in earlier sections can then be used to simultaneously estimate the coefficients of FIR or step-response models from orders M to N.

4.8.1 Simulation Example

The 15^{th} and 30^{th} order FIR coefficients determined by the LS_y and AUDI methods for the 3^{rd} order process described in Section 4.5 are compared in Figure 4.8. Usually the FIR coefficients can be divided into initial and tail-end sections as shown in this figure by the vertical bars. The initial section of the FIR coefficients capture the high frequency and unusual part of the process dynamics whereas the tail-end or low frequency portion is characterized by an exponential decay function corresponding to the first order model [27]. Therefore an optimal approach for estimating FIR models would be to fit the tail-end by an exponentially decaying function and use the AUDI or LS_y methods to identify the initial portion of the FIR coefficients. Such an approach not only avoid the problems of truncation error but it also gives smooth frequency characteristics. Since the tail-end function is a smooth extrapolation of the initial portion of the FIR coefficients, the tail-end function varies according to the order of the identified FIR model to maintain the continuity of the parameter profile.

Figure 4.8 shows that the tail-end portions of the 30^{th} and 15^{th} order FIR models are approximated by the decaying functions $e^{-0.22T}$ and $e^{-0.26T}$ respectively. The vertical bars on the FIR response indicate the point from where the tail-end approximation is made. In this case, the last 5 data points of the initial portion of the FIR parameters (that excludes truncation error) are used to extrapolate the tail-end portion. A significant difference is observed in the extrapolated truncation terms between the 30^{th} and 15^{th} order FIR models because, the 15^{th} order model is not adequate in this case to capture the dominant dynamics of the process.

Figure 4.9 compares the Nyquist plots between the true plant and 15^{th} and 30^{th} order FIR models, where the tail-ends are approximated using the method described above to obtain better frequency characteristics of the model. No significant difference is observed between the models obtained from LS_y and AUDI methods when low-frequency square wave excitation is used as shown in Table 4.10. For a white noise or RBS excitation, a small variance of the order of 10^{-6} is however observed between the FIR parameters obtained using the LS_y and AUDI approaches. The figure also shows that the 15^{th} order FIR model is inadequate to describe this process, which is verified by both the AUDI and LS_y approaches.

The method just described for the FIR model can also be extended to step response model which is illustrated by the above example in Figure 4.10. The extrapolation function used in this case is $0.5[1 - e^{-0.22(t-11)}]$ for the 30^{th} order step response model. Figure 4.10 also shows no apparent difference between the AUDI and LS_y based step response models.

4.9 Application of the AUDI method for estimation of Markov-Laguerre models

The AUDI method described for FIR and Laguerre models can be combined to estimate parameters of the Markov-Laguerre models. Markov-Laguerre models have the advantage of better identifiability in the presence of measurement noise because of the Laguerre component and they are also well suited for processes with significant time delays and non-minimum phase behavior due to the presence of FIR components [15]. Although Laguerre models can be used to represent processes with delays as noted by Dumont *et al.* [10], it is should be observed that a Markov-Laguerre model gives a better representation for a delayed process as also viewed by Finn *et al.* [15]. For a known delay, d, or for a known duration, d, of inverse response and for an upper limit (N) on the order of Laguerre model, the data vector is given by:

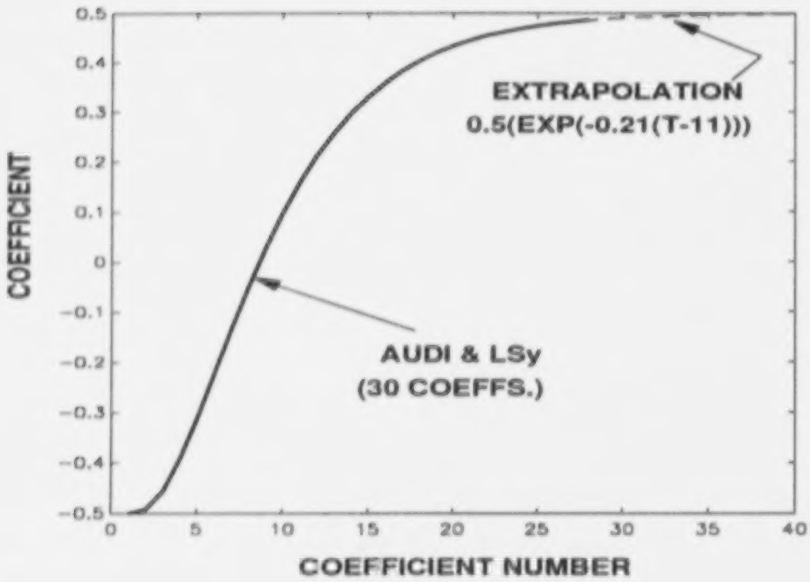


Figure 4.10: A comparison of 30th order step response coefficients obtained using the AUDI and LS_y methods.

$$\phi(t,d,N) = [-y(t), u(t-1), \ldots, u(t-d), l_1(t-d), \ldots, l_N(t-d)]^T \tag{4.70}$$

The AUDI method described in this section can also be used to estimate the Markov-Kautz [15] models, which can be used to approximate the dynamics of underdamped systems with a delay or inverse response.

4.9.1 Simulation Example

The use of Markov-Laguerre model is illustrated for the plant described in Section 4.5 to which a delay of 4 sampling intervals has been added. Nyquist plots between the true plant and the estimated Markov-Laguerre models with 4 FIR coefficients and 3rd and 4th order Laguerre models are compared in Figure 4.11 as obtained by the LS_y and AUDI methods. The Nyquist plots show that the LS_y and AUDI results are in good agreement and they also capture very effectively the high frequency dynamics due to the delay. Figure 4.11 shows that the 4th order Markov-Laguerre model matches very well with the true process.

4.10 Conclusions

- The AUDI method can be used to simultaneously estimate the parameters of 1 to N^{th} order orthonormal function models such as Laguerre models with real or complex

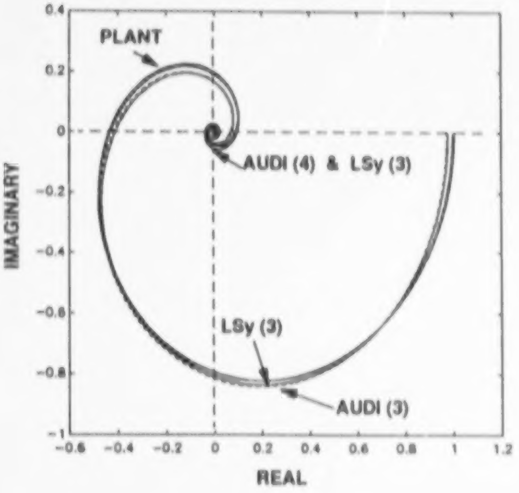


Figure 4.11: A comparison between 3^{rd} and 4^{th} order Markov-Laguerre models obtained using the AUDI and LS_y methods.

poles, FIR (also step-response), Kautz and Markov-Laguerre models.

- LU decomposition of $S^*(t,N)$ gives LS_y solution and it is more robust to process-noise than the AUDI decomposition of $S(t,N)$.

- The solutions obtained by the AUDI method are close to the LS_y solutions. The difference in AUDI and LS_y solutions are due to the use of different objective functions. However the difference between AUDI and LS_y solution diminishes with increase in the model order. This procedure can therefore be used to verify an appropriate order for the AUDI and LS_y models.

- The AUDI method is sensitive to measurement noise. The sensitivity to measurement noise is reduced by replacing the noisy output by a LS_y based model output of suitable model order N' in the data regressor $\phi(t,N)$. The order N' should be greater than the maximum desired order N in the AUDI formulation.

- The method of $\mathrm{UDU^T}$ decomposition is computationally most efficient in calculating the AUDI parameter set $\hat{\Xi}(t,N)$, if the DPMM $(S(t,N))$ matrix is full rank. For rank deficient $S(t,N)$, LU decomposition can be used to calculate $\hat{\Xi}(t,N)$ via the transformation $\mathcal{U}(t,N) = \mathcal{L}^{-T}(t,N)$. The $\mathrm{UDU^T}$ decomposition can be used for rank deficient $S(t,N)$ after suitable modifications in the data regressor as described in *remark*-4.6.

- The truncation error in the FIR model can be reduced by approximating the tail end of the FIR model by an exponential decaying function $e^{-\alpha t}$, where α represents the dominant time constant of the process.

Table 4.2: Laguerre gains obtained using LS_y method.

# of Laguerre filters	Laguerre gains g_i	$\mathcal{J}_y(t)$	$\frac{\mathcal{J}_y(t)}{\sum_{i=1}^{t} y^2(i)}\%$	$\lVert G(\omega) - \hat{G}(\omega)\rVert_2$
1	0.2782	8.2064	5.4365%	3.1351
2	0.2070, 0.0993	0.7454	0.4938%	1.3058
3	0.1779, 0.1519, -0.0431	0.0471	0.0312%	0.3125
4	0.1790, 0.1581, -0.0558, 0.0112	0.0026	0.0017%	0.1579

Table 4.3: Laguerre gains obtained using AUDI method.

# of Laguerre filters	Laguerre gains g_i	$\mathcal{J}_{\hat{y}}(t)$	$\frac{\mathcal{J}_{\hat{y}}(t)}{\sum_{i=1}^{t} y^2(i)}\%$	$\lVert G(\omega) - \hat{G}(\omega)\rVert_2$
1	0.2942	8.6782	5.7491%	3.2450
2	0.1999, 0.1092	0.8199	0.5432%	1.1921
3	0.1760, 0.1554, -0.0460	0.0503	0.0333%	0.2838
4	0.1791, 0.1585, -0.0565, 0.0118]	0.0028	0.0018%	0.1617

Table 4.4: Comparison of FIR coefficients obtained using LS_y and AUDI methods.

#	AUDI	LS_y	#	AUDI	LS_y	#	AUDI	LS_y
1	0.0066	0.0061	11	0.0539	0.0539	21	0.0095	0.0095
2	0.0358	0.0358	12	0.0465	0.0465	22	0.0079	0.0079
3	0.0630	0.0630	13	0.0397	0.0397	23	0.0065	0.0065
4	0.0802	0.0802	14	0.0337	0.0337	24	0.0054	0.0054
5	0.0879	0.0879	15	0.0284	0.0284	25	0.0044	0.0044
6	0.0885	0.0885	16	0.0239	0.0239	26	0.0037	0.0037
7	0.0845	0.0845	17	0.0200	0.0200	27	0.0030	0.0030
8	0.0779	0.0779	18	0.0167	0.0167	28	0.0025	0.0025
9	0.0701	0.0701	19	0.0139	0.0139	29	-0.012	0.0020
10	0.0619	0.0619	20	0.0115	0.0115	30	0.0223	0.0075

Appendix-1: UDU^T Decomposition

This appendix shows that the matrix $\mathcal{U}(t,N)$ contains the AUDI parameter set $\hat{\Xi}(t,N)$ which is obtained via the transformation $\mathcal{U}(t,N) = \mathcal{L}^{-T}(t,N)$. This is illustrated through an example for $N = 3$ and by considering the matrices L_k and $\mathcal{L}(t,N)$ in equations (4.44) and (4.45). For the sake of simplicity in the notations, $\hat{\xi}(t,N)$ is written as $\hat{\xi}(N)$ (*i.e.* by dropping the argument (t)) in the following example:

$$
\begin{aligned}
&\mathcal{L}(t,3) = [L_3 L_2 L_1] = \\
&\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\hat{\xi}_1(3) & -\hat{\xi}_2(3) & -\hat{\xi}_3(3) & 1 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\hat{\xi}_1(2) & -\hat{\xi}_2(2) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\xi_1(1) & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
&- \begin{pmatrix} 1 & 0 & 0 & 0 \\ \hat{\xi}_1(1) & 1 & 0 & 0 \\ \hat{\xi}_1(2) + \hat{\xi}_2(2)\hat{\xi}_1(1) & \hat{\xi}_2(2) & 1 & 0 \\ \hat{\xi}_1(3) + \hat{\xi}_2(3)\hat{\xi}_1(1) + \hat{\xi}_3(3)\hat{\xi}_1(2) + \hat{\xi}_3(3)\hat{\xi}_2(2)\hat{\xi}_1(1) & \hat{\xi}_2(3) + \hat{\xi}_3(3)\hat{\xi}_2(2) & \hat{\xi}_3(3) & 1 \end{pmatrix}
\end{aligned}
\tag{4.71}
$$

then

$$
\begin{aligned}
\mathcal{U}(t,3) &= [\mathcal{L}(t,3)]^{-T} \\
&= \begin{pmatrix} 1 & \hat{\xi}_1(1) & \hat{\xi}_1(2) & \hat{\xi}_1(3) \\ 0 & 1 & \hat{\xi}_2(2) & \hat{\xi}_2(3) \\ 0 & 0 & 1 & \hat{\xi}_3(3) \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}
\tag{4.72}
$$

Appendix-2: Solution for $\xi(t,N)$ and $\theta(t,N)$ parameters.

The purpose of this example is to show that the parameter vector $\theta(t,N)$ is equivalent to $\xi(t,N)$ for a *complete* system *i.e.* when the model is equal to the plant and where the plant is noise free. This is illustrated using symbolic computation for a 3^{rd} order orthonormal model. The argument t is dropped from $\theta(t,N)$, $\xi(t,N)$ and other intermediate variables in this example for the sake of simplicity. Consider the following 3^{rd} order orthonormal model:

$$
y(t) = h^T(t,3)\theta(3) \tag{4.73}
$$

where $\theta(3) = [g_1, g_2, g_3]^T$ and the regressor vector is: $h(t,3) = [\psi_1(t), \psi_2(t), \psi_3(t)]^T$. For any 3 points $Y(t,3) = [y(t-2), y(t-1), y(t)]^T$, $\theta(3)$ is given by:

$$\theta^{(3)} = H^{-1}(t,3)Y(t,3) \tag{4.74}$$

where $H(t,3) = [h(t-2,3), h(t-1,3), h(t,3)]^T$. The following solution (4.74) is then obtained using Matlab [11] symbolic computation:

$$\theta(3) = [g_1, g_2, g_3]^T = \frac{1}{d}[a_1, -a_2, a_3]^T \tag{4.75}$$

where

$$\begin{aligned}
d &= \psi_1(t-2)\psi_2(t-1)\psi_3(t) - \psi_1(t-2)\psi_2(t)\psi_3(t-1) - \\
&\quad \psi_1(t)\psi_2(t-1)\psi_3(t-2) - \psi_1(t-1)\psi_2(t-2)\psi_3(t) + \\
&\quad \psi_1(t-1)\psi_2(t)\psi_3(t-2) + \psi_1(t)\psi_2(t-2)\psi_3(t-1) \\
a_1 &= y(t-2)[\psi_2(t-1)\psi_3(t) - \psi_2(t)\psi_3(t-1)] + \\
&\quad y(t-1)[\psi_3(t-2)\psi_2(t) - \psi_2(t-2)\psi_3(t)] + \\
&\quad y(t)[\psi_2(t-2)\psi_3(t-1) - \psi_2(t-1)\psi_3(t-2)] \\
a_2 &= y(t-2)[\psi_1(t-1)\psi_3(t) - \psi_1(t)\psi_3(t-1)] + \\
&\quad y(t-1)[\psi_1(t)\psi_3(t-2) - \psi_1(t-2)\psi_3(t)] + \\
&\quad y(t)[\psi_1(t-2)\psi_3(t-1) - \psi_1(t-1)\psi_3(t-2)] \\
a_3 &= y(t-2)[\psi_1(t-1)\psi_2(t) - \psi_1(t)\psi_2(t-1)] + \\
&\quad y(t-1)[\psi_1(t)\psi_2(t-2) - \psi_1(t-2)\psi_2(t)] + \\
&\quad y(t)[\psi_1(t-2)\psi_2(t-1) - \psi_1(t-1)\psi_2(t-2)]
\end{aligned} \tag{4.76}$$

Similarly, according to the AUDI approach, equation (4.73) can be rewritten as:

$$\psi_3(t) = -\phi^T(t,3)\xi(3) \tag{4.77}$$

again using symbolic computation it can be shown that the solution for $\xi(3)$ is:

$$\xi(3) = \left[\frac{d}{a_3}, -\frac{a_1}{a_3}, -\frac{a_2}{a_3}\right]^T \tag{4.78}$$

Comparing solutions (4.75) and (4.78) it can be inferred that they are equivalent, *i.e.* when $\xi(3)$ is converted into LS_y form, it exactly results into $\theta(3)$ as shown by equation (4.75).

---

[11] The Math Works Inc.

Bibliography

[1] Y.W. Lee. *Statistical Theory of Communication.* Wiley, New York, USA, 1960.

[2] T.Y. Young and W.H. Huggins. "Complementary Signals and Orthogonalized Exponentials". *IEEE Trans. Circuit Theory*, 9:362 –370, 1962.

[3] R.E. King and P.N. Paraskevopoulos. "Digital Laguerre Filters". *Circuit Theory and Applications*, 5:81 – 91, 1977.

[4] A.C. Brinker. "Adaptive Orthonormal Filters". In *IFAC World Congress*, volume 5, pages 287 – 292, Sydney, Australia, 1993.

[5] P. Beckmann. *Orthogonal Polynomials for Engineers and Physicists.* The Golem Press, Boulder, Colorado, USA, 1973.

[6] Y. Fu and G.A. Dumont. "An Optimal Time Scale for Discrete Laguerre Network". *IEEE Trans. on AC*, 38(6):934 – 938, 1993.

[7] W.R. Cluett and L. Wang. "Modelling and Robust Controller Design using Step Response Data". *Chem. Eng. Sci.*, 46:2065 –2077, 1991.

[8] L. Wang and W.R. Cluett. "Optimal Choice of Time-scaling Factor for Linear Approximations using Laguerre Models". *IEEE AC*, 1995. To appear.

[9] B. Wahlberg. "System Identification Using Laguerre Models". *IEEE Trans. on AC*, 36(5):551 – 562, 1991.

[10] G.A. Dumont, A. Elnaggar, and A. Elshafei. "Adaptive Predictive Control of Systems with Time-Varying Time Delay". *Int. J. of Adaptive Control and Signal Processing*, 7(6):91 – 101, 1993.

[11] P. Lindskog and B. Wahlberg. "Applications of Kautz Models in System Identification". In *12th IFAC World Congress*, volume 5, pages 309 – 312, Sydney, Australia, July 18-23, 1993.

[12] G.Y. Dumont, C.C. Zevros, and G.L. Pageau. "Laguerre-based Adaptive Control of pH in an Industrial Bleach Plant Extraction Stage". *Automatica*, 26(4):781 – 787, 1990.

[13] B.J. Allison, G.A. Dumont, L.H. Novak, and W.J. Cheetam. "Adaptive-Predictive Control of Kamyr Digester Chip level". *AIChEJ*, 36(7):1075 – 1086, 1990.

[14] W.R. Cluett and L. Wang. "Frequency Smoothing using Laguerre model". *IEE Proceedings-D*, 139(1):88 – 96, 1992.

[15] C.K. Finn, B. Wahlberg, and B.E. Ydstie. "Constrained Predictive Control Using Orthonormal Expansions". *AIChEJ*, 39(11):1810 – 1826, 1993.

[16] S. Niu, D. Xiao, and D.G. Fisher. "A Recursive algorithm for Simultaneous Identification of Model Order and Parameters". *IEEE Trans. on ASSP*, 38(5):884 – 889, 1990.

[17] S. Niu, D.G. Fisher, and D. Xiao. "An Augmented UD Identification Algorithm". *Int. J. Control*, 56(1):193 – 211, 1991.

[18] G Dahlquist and A. Björck. *Numerical Methods.* Prentice Hall, New Jersey, USA, 1974. Translated by N.Anderson.

[19] G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation.* Academic Press, New York, USA, 1977.

[20] S. Niu, L. Ljung, and A. Björck. "Decomposition methods for Least-Squares Parameter Estimation". *Accepted for publication in IEEE Trans. on SP*, 1995.

[21] J.T. Oden. *Applied Functional Analysis.* Prentice Hall, New Jersey, USA, 1979.

[22] E. Kreyszig. *Introductory Functional Analysis with Applications.* John Wiley and Sons, USA, 1978.

[23] M. Laszlo. *Hilbert Space methods in Science ad Engineering.* Adam Higler, Bristol, U.K. and Akademiai Kiado, Budapest, Hungary., 1989.

[24] G. Sansone, A.H. Diamond, and E. Hille. *Orthogonal Functions.* Interscience Publishers Inc., New York, USA, 1959.

[25] G. Dumont. "Laguerre Network models for Predictive Control". In *IEEE CAC Workshop on Model Predictive Control*, Vancouver, Canada, December 13, 1993. Ed: S.L.Shah and G.Dumont.

[26] S. Niu. *Augmented UD Identification for Process Control.* Ph.D. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1994.

[27] K.Y. Kwok and S.L. Shah. "Long-Range Predictive Control with a Terminal Matching Condition". *Chem. Eng. Sci.*, 49(9):1287 – 1300, 1994.

Chapter 5

Robust Design of Generalized Predictive Control (GPC).

The small-gain theorem is used to obtain robust design guidelines for the GPC control law. The effect of different tuning parameters on the robustness of GPC is examined via analytical methods, simulations and experimental evaluations. This method relies on *a priori* knowledge of the model-plant mismatch which can be estimated from either open loop or closed loop plant data. The proposed robust design method can also be used to select some of the GPC tuning parameters using parameter optimization techniques.

5.1 Introduction

The performance and robustness of a model based controller to a large extend depends on how well a model is able to capture the dynamics of a plant. A mathematical model can have several degrees of complexity, but invariably in a realistic situation a model cannot exactly emulate a physical process, and the problems of stability and performance in a system mostly manifest themselves from this model-plant uncertainty. Furthermore for most real-time control applications, simpler models such as linear input-output transfer functions are desirable. But the performance and stability limitations are often significant for such linear models owing to the constraints on model order and structure. Identification methods discussed in Chapter 3 can be used to estimate a linear model of the plant.

One of the important robustness tools that can be used examine the stability of a system in the presence of model-plant uncertainty is the norm bounded ***small-gain theorem*** (SGT) based on the Nyquist stability criterion [1, 2]. The small gain formulation is only a

---

[1] **Material from this chapter has been published as: (a) Banerjee,P. and S.L.Shah, 'The Role of Signal Processing Methods in the Robust Design of Predictive Control', *Automatica*, vol 31, No. 5, pp 681-695, 1995 and (b) Banerjee,P. and S.L.Shah, 'Robust Stability of GPC as Applied to a First order Model with Delay', Proc. of ACC, Vol 999, Pg. 999-999, Seattle, 1995.**

sufficient condition for stability, which makes the design of a controller potentially conservative. The key difficulty in the application of SGT is the knowledge of model uncertainty bounds. Inadequate estimation of the uncertainty bound can lead to instability whereas a conservative bound can reduce the performance of a system. The quantification of such uncertainty bounds is currently an active area of research [3, 4, 5, 6, 7, 8, 9]. Several parametric and nonparametric approaches have been proposed to estimate the error bounds. One such method is due to Kosut (1987) [3], which uses the discrete Fourier transform (DFT) of the output error to estimate the spectrum of the model plant mismatch (MPM). Kosut's results indicate a good fit of the uncertainty over a certain frequency band for the noise free case. However in Chapter 2 it was shown that, with the use of classical signal processing smoothing techniques [10, 11, 12], it is possible to obtain a good estimate of the uncertainty over the entire frequency range of interest for the robustness analysis.

An important class of controllers that has received widespread acceptance and success in the chemical process industry is model-based, long range predictive controller (LRPC) [13, 14, 15, 16]. To fully exploit the advantages offered by such controllers, it becomes meaningful to develop robust design guidelines for successful implementation of such controllers in industry. Robust design of linear quadratic (LQ) optimal controllers within SGT framework is discussed in Bitmead *et al.* (1990) [17]. Robust design of an observer prefilter for one such LQ controller, namely the ***generalized predictive control*** (GPC) [14] using the SGT tool was examined by Robinson and Clarke (1991) for mean-level and dead-beat performance [18]. This chapter examines the tuning parameters of GPC from the perspective of robustness analysis. The robustness properties of some of these tuning parameters are also analytically established in this chapter.

Analytical derivations of robustness measures involve development of explicit expressions for the GPC Diophantine coefficients and for the polynomials that represent GPC in the feedback form. These expressions in themselves are an important contribution because they: (a) can increase the computational speed of the GPC algorithm and (b) have potential use in adaptive GPC control. The use of Maple[2] (a symbolic mathprocessor), proved to be indispensable in deriving expressions for the GPC Diophantine coefficients. The analytical results presented in this chapter are limited to first order models[3] as it will become clear that a generic proof for the robust stability of GPC when applied to models of order higher than one becomes a nontrivial problem. To the best of this author's knowledge, this is the first complete and systematic application study where the robust design techniques for GPC are developed by simulations, experimental evaluations and supported analytically (for some of the key tuning parameters of GPC). This design method also helps to select appropriate tuning parameters such that the controller gives 'reasonably good'[4] performance but with guaranteed robustness to the estimared MPM. This work is also extended to the robust design of GPC in a quasi-adaptive framework by using closed-loop data to estimate the MPM.

---

[2] Maple is a registered trademark of Symbolic Computation Group, Waterloo Maple Software, University of Waterloo, Canada, 1981-1990.

[3] Any overdamped process can be approximated by a first order model with a delay [19]. Moreover in the chemical industry, overdamped processes are quite common and hence these results can be applied.

This chapter is organized as follows: (a) The principle behind GPC is briefly discussed in Section 5.2, (b) The expressions for the GPC Diophantine coefficients and linear form of GPC are derived in Sections 5.3 and 5.4, (c) Section 5.5 describes the formulation of GPC in an SGT framework for robustness analysis, (d) The effect of different GPC tuning parameters is discussed in Sections 5.7 to 5.12, (e) The effects of model quality and noise/disturbances on the robustness of GPC are discussed in Sections 5.13 and 5.14, (f) The proposed robust tuning guidelines for GPC are evaluated experimentally in Section 5.15, (g) Estimation of modelling errors and the model from closed loop data are presented in Section 5.16, (h) Use of an optimization technique for the selection of appropriate GPC tuning parameters is discussed in Section 5.17 and (i) The chapter ends with concluding remarks in Section 5.18.

5.2 Generalized predictive Control

Most LRPC strategies (including GPC) are based on the minimization of a multistep quadratic cost function J, such as:

$$J = \sum_{i=N_1}^{N_2} [w(t+1) - \hat{y}(t+i|t)]^2 + \sum_{i=1}^{NU} \lambda[\Delta u(t+i-1)]^2 \tag{5.1}$$

where N_1, N_2, NU and λ are the tuning parameters representing the minimum and maximum prediction (or costing) horizon, the control horizon and control weighting (or move supression factor) respectively. In equation-(5.1), $w(t+i)$ and $\hat{y}(t+i|t)$ are the projected set-points and predicted outputs over the prediction horizon and $\Delta u(t+i-1)$ represents incremental controller moves over the control horizon. The control law is obtained by minimizing J with respect to $\Delta u(t+i-1)$. The predicted output $\hat{y}(t+i|t)$ is model dependent and in the case of GPC, this model has an ARIMAX structure.

In order to improve the controller performance, the objective function J in equation-(5.1) can be modified to include: (a) weightings on the predicted output, (b) weightings on the controller movments and (b) steady-state weighting [20, 21, 22, 23]. Some of these weightings in the objective function are useful for control-relevant identification, which is discussed in Chapter 7. Recently it has been shown that the objective function J given by equation-(5.1) can be modified to: (a) have variable spacings in the prediction horizon and (b) use interpolation functions in J [24], but such modifications are not considered here. This chapter pertains only to the following weighted objective function for the GPC [20, 23]:

---

[4] The definition of a good performance depends a lot on the plant dynamics. A controller that gives a dead beat response without violating any constraints can be said to define the characteristics of the best output performance that can be achieved. This usually is not possible to achieve in practice because of modelling errors and plants (particularly for chemical processes) with high orders and which are often characterized by the time delays, inverse response or other unusual high frequency dynamics. Nevertheless, a good or an acceptable performance can be said to have zero offset, a minimum overshoot with a response or settling time much faster than the mean level performance and the controller actions are smooth and within the constraints.

$$J = \sum_{i=N_1}^{N_2} \Gamma_y[w(t+i) - P(q)\hat{y}(t+i|t)]^2 + \sum_{i=1}^{NU} \lambda[\Delta u(t+i-1)]^2 + \gamma_\infty[\hat{y}_{ss} - w_{ss}]^2 \tag{5.2}$$

where $\Gamma_y = \mathrm{diag}[\gamma_y(N_1), \gamma_y(N_1+1), \cdots, \gamma_y(N_2)]$ is the output weighting, γ_∞ is the steady-state weighting, $\hat{y}_{ss}$ is output at the steady-state, w_{ss} is the set-point at ∞-horizon and $P(q)$ can either be a transfer function or a polynomial that imparts model following characteristics to the controller. In this chapter $P(q)$ is considered to be a polynomial. J in equation-(5.2) can be written more compactly as:

$$J = [\mathbf{w} - \Psi]^T \Gamma [\mathbf{w} - \Psi] + \Delta\mathbf{u}^T \Lambda \Delta\mathbf{u} \tag{5.3}$$

where:

$$\begin{aligned}
\Gamma &= \mathrm{diag}[\gamma_y(N_1), \gamma_y(N_1+1), \ldots, \gamma_y(N_2), \gamma_\infty] \\
\mathbf{w} &= [w(N_1), w(N_1+1), \ldots, w(N_2), w_{ss}]^T \\
\Psi &= P(q)[[\hat{y}(t+N_1|t), \hat{y}(t+N_1+1|t), \ldots, \hat{y}(t+N_2|t)], \hat{y}_{ss}]^T \\
&= [\Psi(t+N_1|t), \Psi(t+N_1+1|t), \ldots, \Psi(t+N_2|t), \Psi_{ss}]^T \\
\Lambda &= \mathrm{diag}[\lambda(1), \lambda(2), \ldots, \lambda(NU)] \\
\Delta\mathbf{u} &= [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+NU-1)]^T
\end{aligned} \tag{5.4}$$

such that $P(1) = 1$ and all values of $\gamma_y(i)$ in Γ are identical to γ_y, $\lambda(i)$ in Λ are identical to λ and $\Psi_{ss} = \hat{y}_{ss}$.

Note 5.1 *As in the case of previous chapters, all polynomials in this chapters are in terms of q^{-1}, although for the sake of convenience, q is used in the arguments.*

The unconstrained minimization of J in equation-(5.3) with respect to $\Delta\mathbf{u}$ results in the following projected (GPC) control law [14]:

$$\Delta\mathbf{u} = \underbrace{[\mathcal{G}^T\mathcal{G} + \Lambda]^{-1}\mathcal{G}^T}_{\mathbf{h}}[\mathbf{w} - \mathbf{f}] \tag{5.5}$$

where $\mathcal{G}$ is the step-response matrix and $\mathbf{f}$ is the free-response vector whose derivation is shown in the following section.

5.3 Diophantine equations

For the ARIMAX process-model:

$$A(q)y(t) = B^{\star}(q)u(t-1) + \frac{C(q)}{\Delta}\xi(t) \tag{5.6}$$

the projected output $P(q)\hat{y}(t+i|t)$ for GPC in the equation-(5.2), is obtained by multiplying equation-(5.6) by $P(q)$ and taking the Diophantine identity[5] of the resultant noise model as:

$$\frac{P(q)C(q)}{\Delta A(q)} = E_i(q) + q^{-i}\frac{F_i(q)}{\Delta A(q)} \tag{5.7}$$

where $A(q)$, $B^{\star}(q)$, $C(q)$ are polynomials as mentioned in Chapter 3, Δ is the difference operator $1-q^{-1}$ and $E_i(q)$ and $F_i(q)$ are the i^{th} quotient and remainder of $P(q)C(q)/[A(q)\Delta]$.

Note 5.2 *The numerator polynomial $B^{\star}(q)$ includes only the process delays and excludes the zero order hold (ZOH). The ZOH is considered by taking the product $B^{\star}(q)u(t-1)$ as shown by equation-(5.6). The numerator polynomial is designated by $B(q)$ if it includes the ZOH i.e. $B(q) = q^{-1}B^{\star}(q)$.*

For the degree of a polynomial denoted by $\delta[\cdot]$, we then define $na = \delta[A(q)]$, $nb = \delta[B^{\star}(q)]$ and $nc = \delta[C(q)]$. Combining equations-(5.6) and (5.7) results in:

$$\hat{y}(t+i|t) = \frac{E_i(q)B^{\star}(q)}{C(q)}\Delta u(t+i-1) + \frac{F_i(q)}{C(q)}y(t) \tag{5.8}$$

The expression $E_iB^{\star}(q)/C(q)$ in equation-(5.8) can be further factored as:

$$\frac{E_iB^{\star}(q)}{C(q)} = \mathcal{G}_i(q) + q^{-i}\frac{\bar{G}_i(q)}{C(q)} \tag{5.9}$$

which upon substitution in equation-(5.8) results in:

$$\Psi(t+i|t) = \mathcal{G}_i(q)\Delta u(t+i-1) + f(t+i) \tag{5.10}$$

where $f(t+i)$ is the filtered free-response, which is given by:

$$f(t+i) = \bar{G}_i(q)\Delta u^{f}(t-1) + F_i(q)y^{f}(t) \tag{5.11}$$

where $y^{f}(t) = y(t)/C(q)$ and $\Delta u^{f}(t-1) = \Delta u(t-1)/C(q)$. For the prediction horizon N_1 to N_2, equation-(5.11) can be compactly expressed as:

---

[5] Recently a new approach for the recursion of the Diophantine equations has emerged for GPC [24]. However the classical Diophantine approach is pursued in this thesis because it adequately serves the purpose of deriving various proofs in this chapter. However as a future research direction, it would be of interest to explore the effectiveness of the new Diophantine approach to solve the problems outlined in this chapter.

$$\mathbf{f} = \bar{\mathbf{G}}\Delta u^f(t-1) + \mathbf{F}y^f(t) \tag{5.12}$$

for use in equation-(5.5). In reality the noise polynomial $C(q)$ is unknown, hence it is replaced by its estimated value $C_c(q)$ $(nc = \delta(C_c(q)))$ which then becomes an important tuning parameter of GPC. The $C_c(q)$ filter is more popularly known as the T-filter or the observer polynomial and helps in rejecting undesirable effects of the unmodelled dynamics. At steady-state the free-response is given by [22]:

$$f_s = \bar{G}_s(q)\Delta u^f(t-1) + F_s(q)y^f(t) \tag{5.13}$$

where

$$\begin{aligned} F_s(q) &= \frac{C_c(1)}{A(1)}A(q) \\ \bar{G}_s(q) &= \frac{1}{A(1)\Delta}[B^\star(1)C_c(q) - C_c(1)B^\star(q)] \end{aligned} \tag{5.14}$$

In order to analytically establish robustness properties of the GPC tuning parameters, it is necessary to express the Diophantine polynomials $E_i(q)$, $F_i(q)$, $G_i(q)$ and $\mathcal{G}_i(q)$ as functions of the GPC tuning knobs and the process model. In this context, first consider the denominator polynomial:

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_{na}q^{-na} \tag{5.15}$$

whose roots are $r_i \in \Im^{na}$. Using Maple, it is found that the coefficients a_i in equation-(5.15) are related to the roots r_i in the following way:

$$\begin{aligned} a_1 &= -(r_1 + r_2 + \cdots + r_{na}) \\ a_2 &= r_1(r_2 + \cdots + r_{na}) + r_2(r_1 + r_3 + \cdots + r_{na}) + \cdots \\ &\qquad +r_{na}(r_1 + r_2 + \cdots + r_{na-1}) \\ a_3 &= -[r_1r_2(r_3 + \cdots + r_{na}) + r_2r_3(r_1 + r_4 + \cdots + r_{na}) + \cdots \\ &\qquad +r_{na-1}r_{na}(r_1 + \cdots + r_{na-2})] \\ &\vdots \\ a_{na} &= (-1)^{na}(r_1r_2\ldots r_{na}) \end{aligned} \tag{5.16}$$

From the above equation-(5.16) it is clear that the expression for the coefficients a_i becomes a complicated function of the roots r_i as the value of na increases. Hence:

Note 5.3 *For the sake of simplicity in the following analytical exposition, the model is restricted to first order i.e.* $na = 1$ *where* $a = -r$.

The justification for the above note-5.3 will become more apparent with further progress in the analytical developments presented later in this chapter. Analytical derivations for the Diophantine polynomials based on a first order model are considered next in the subections-5.3.1 to 5.3.4.

5.3.1 The quotient polynomial $E_i(q)$ in the Diophantine identity

The factor $1/\Delta A(q)$ for $na = 1$ (for $0 < a < 1$) in equation-(5.7) can be expressed as:

$$\frac{1}{\Delta A(q)} = 1 + (1+a)q^{-1} + (1+a+a^2)q^{-2} + \cdots + (1+a+a^2+\cdots+a^{\infty})q^{-\infty} \quad (5.17)$$

If the geometric series in equation-(5.17) is denoted by:

$$1 + a + a^2 + a^3 + \cdots + a^n = \frac{1-a^{n+1}}{1-a} = \mathcal{P}_1(n+1) \quad (5.18)$$

then equation-(5.17) can be rewritten as:

$$\frac{1}{\Delta A(q)} = \mathcal{P}_1(1) + \mathcal{P}_1(2)q^{-1} + \mathcal{P}_1(3)q^{-2} + \cdots + \mathcal{P}_1(\infty)q^{-\infty} \quad (5.19)$$

Assuming $P(q) = 1$ and $C(q) = 1$, then from equations-(5.7), (5.17) and (5.19), it is shown that:

$$\begin{aligned} E_i(q) &= 1 + e_1 q^{-1} + e_2 q^{-2} + \cdots + e_i q^{-i} \\ &= 1 + \mathcal{P}_1(2)q^{-1} + \mathcal{P}_1(3)q^{-2} + \cdots + \mathcal{P}_1(i+1)q^{-i} \end{aligned} \quad (5.20)$$

For $C(q) = 1 - cq^{-1}$, e_i in equation-(5.20) becomes:

$$\begin{aligned} e_i &= \frac{1}{1-a}[(1-a^{i+1}) - c(1-a^i)] \\ &= \mathcal{P}_1(i+1) - c\mathcal{P}_1(i) \end{aligned} \quad (5.21)$$

5.3.2 The remainder polynomial $F_i(q)$ in the Diophantine identity

Lemma 5.1 *The Diophantine polynomial $F_i(q)$ in equation-(5.7) holds the following relation [21]:*

$$q^{-1}F_i(q) = F_{i-1}(q) + e_{i-1}q^{-1}\Delta A(q) \quad (5.22)$$

Proof: Rearranging the Diophantine identity in equation-(5.7) yields:

$$q^{-i}F_i(q) = 1 - E_j(q)\Delta A(q) \quad (5.23)$$

similarly for $i-1$, we have:

$$q^{-(i-1)}F_{i-1}(q) = 1 - E_{i-1}(q)\Delta A(q) \tag{5.24}$$

Substituting equation-(5.20) in equations-(5.23) and (5.24) and subtracting the resulting equations, proves the lemma.

A combination of *lemma*-5.1 and equation-(5.18) for $C(q) = 1 - cq^{-1}$ gives the following expression for $F_i(q)$:

$$\begin{aligned} F_i(q) &= \frac{1}{1-a}[1 - a^{i+1} - c(1-a^i) \qquad - a(1 - a^i - c(1 - a^{i-1}))] \\ &= \frac{1}{1-a}[(1-c) + a^i(c-a) \qquad - a((1-c) + a^{i-1}(c-a))] \\ &= [\mathcal{P}_1(i+1) - c\mathcal{P}_1(i) \qquad -a(\mathcal{P}_1(i) - c\mathcal{P}_1(i-1))] \\ &= [f_{i1} \qquad f_{i2}] \end{aligned} \tag{5.25}$$

Note 5.4 *The above expression (equation-(5.25)) becomes difficult to generalize for na ≥ 2, hence this is an additional reason to restrict the analysis presented in this chapter to a first order model.*

5.3.3 The quotient polynomial $\mathcal{G}_i(q)$ in the Diophantine identity

For the model numerator in equation-(5.6) (after considering the ZOH):

$$B(q) = q^{-d}(b_1q^{-1} + b_2q^{-2} + \cdots + b_{nb}q^{-nb}) \tag{5.26}$$

(where d is the delay), using Maple and equations-(5.9) and (5.20), the Diophantine polynomial $\mathcal{G}_i(q)$ is shown to be (where $i \in [N_1, N_2]$ and $N_2 > \max(d, nb)$):

$$\begin{aligned} \mathcal{G}_i(q) &= g_{d+1}q^{-(d+1)} + g_{d+2}q^{-(d+2)} + \cdots + g_{d+nb}q^{-(d+nb)} + \cdots + g_{d+i}q^{-i} \\ &= b_1q^{-(d+1)} + (b_2 + b_1e_1)q^{-(d+2)} + \cdots + (b_{nb} + b_{nb-1}e_1 + \cdots + b_1e_{nb-1})q^{-(nb+d)} + \cdots \\ &\quad +(b_{nb}e_{i-nb} + b_{nb-1}e_{i-nb+1} + \cdots + b_1e_{i-N_1})q^{-i} \end{aligned} \tag{5.27}$$

where $e_j = 0$ for $j < 0$ and $g_j = 0$ for $j \leq d$. For a first order model of the form: $bq^{-(d+1)}/(1 - a^q{-}1)$, g_i in equation-(5.27) reduces to:

$$\begin{aligned} g_i &= \frac{b(1-a^{i-d})}{1-a} && i > d \\ &= b\mathcal{P}_1(i-d) \\ &= 0 && i \leq d \end{aligned} \tag{5.28}$$

Equation-(5.28) is valid irrespective of $C_c(q)$.

5.3.4 The remainder polynomial $\bar{G}_i(q)$ in the Diophantine identity

$\bar{G}_i(q)$ in equation-(5.9) is given by:

$$\bar{G}_i(q) = \bar{g}_1 + \bar{g}_2 q^{-1} + \cdots + \bar{g}_i q^{-i} \tag{5.29}$$

The expression for $\bar{G}_i(q)$ becomes complicated if nb exceeds 1, therefore:

Note 5.5 *For the sake of simplicity, further analysis is carried out for* $B(q) = bq^{-(d+1)}$.

In view of the above note, from equations-(5.9), (5.21) and (5.18), $\bar{G}(q)$ is given by:

$$\bar{G} = b \begin{bmatrix} & & & & & \mathcal{P}_1(1) \\ & & & & \mathcal{P}_1(1) & e_1 \\ & & & \mathcal{P}_1(1) & e_1 & e_2 \\ & & & \vdots & \vdots & \vdots \\ & & \vdots & \vdots & \vdots & \vdots \\ & \mathcal{P}_1(1) & & & & \\ \mathcal{P}_1(1) & e_1 & & & & \\ \mathcal{P}_1(2) & e_2 & & & & \\ \vdots & \vdots & & & & \vdots \\ \mathcal{P}_1(N_d-1) & e_{N_2-d} & \cdots & \cdots & e_{N_2-2} & e_{N_2-1} \end{bmatrix}_{N_2 \times d}$$

where $N_1 = 1$ in the above equation and in general, $\delta[\bar{G}(q)] = \bar{ng} = \max(nb-1, d-1, nc-1)$ [25].

5.4 Linear form of GPC

Model predictive controllers (MPC) are sometimes referred as open-loop controllers because of the nature of objective function as shown by equation-(5.1) or (5.2). However as a result of repeated implementation of the controller using a receding horizon[6] strategy, the MPC gives the effect of conventional feedback. Therefore the GPC control law in equation-(5.5) can be arranged to a feedback form as:

$$T(q)\Delta u(t) = R(q)w(t) - S(q)y(t) \tag{5.30}$$

where the coefficient polynomials in the above equation are given by [25, 26]:

---

[6] Under the receding horizon strategy, only the first controller move out of the NU controller values calculated over the controller horizon is implemented.

$$\begin{aligned}
R(q) &= C_c(q)\sum_{i=N_1}^{N_2} h_i \\
T(q) &= C_c(q) + q^{-1}[\sum_{i=N_1}^{N_2} h_i\bar{G}_i(q)] \\
S(q) &= \sum_{i=N_1}^{N_2} h_i F_i(q)
\end{aligned} \tag{5.31}$$

and where $h_i, i = [N_1, N_2]$ are elements of the first row of the matrix $[\mathcal{G}^T\mathcal{G} + \mathbf{\Lambda}]^{-1}\mathcal{G}^T$ as shown by equation-(5.5). The polynomials in equation-(5.31) have the following degrees: $\delta[R(q)] = nc$, $\delta[S(q)] = na$ and $\delta[T(q)] = \max(nb, nc) = nt$.

5.4.1 The control law matrix h

With $N_1 = 1$, $NU = 1$, $\gamma_\infty = 0$, $\gamma_y = 1$ and for the model:

$$\hat{G}(q) = \frac{bq^{-(d+1)}}{(1 - aq^{-1})} \tag{5.32}$$

h is given by:

$$\mathbf{h} = [\sum_{i=1}^{N_2} x_i^2 + \lambda]^{-1}[x_1, x_2, \ldots, x_{N_2}] \tag{5.33}$$

where x_i is the i^{th} step-response (which is same as g_is in equation-(5.28)) given by $x_i = b(1 - a^{i-d})/(1 - a)$ and $x_i = 0$ for $i < d$; or from equation-(5.28) x_i can be expressed as:

$$x_i = b\mathcal{P}_1(i - d) \tag{5.34}$$

Substituting equations-(5.34) and (5.18) into equation-(5.33) and taking $d = 0$ results in:

$$\mathbf{h} = \frac{1}{\kappa}\left(\frac{[(1-a), (1-a^2), \ldots, (1-a^{N_2})]}{(1-a)^2 + (1-a^2)^2 + \cdots + (1-a^{N_2})^2 + \kappa^{-2}\lambda}\right) \tag{5.35}$$

Using the summation of series:

$$(a^2)^0 + (a^2)^1 + \cdots + (a^2)^n = \frac{1 - (a^2)^{n+1}}{1 - a^2} = \mathcal{P}_2(n + 1) \tag{5.36}$$

and equation-(5.18) in equation-(5.33) gives:

$$\mathbf{h} = \frac{1}{\kappa}\left(\frac{[(1-a), (1-a^2), \ldots, (1-a^{N_2})]}{N_2 + a^2\mathcal{P}_2(N_2) - 2a\mathcal{P}_1(N_2) + \kappa^{-2}\lambda}\right) \tag{5.37}$$

For a delay of d units, the above relation is generalized to:

$$\mathbf{h} = \frac{1}{\kappa}\left(\frac{[\overbrace{0,0,\ldots,0}^{d\text{-zeros}},(1-a),(1-a^2),\ldots,(1-a^{N_2-d})]}{N_2-d+a^2\mathcal{P}_2(N_2-d)-2a\mathcal{P}_1(N_2-d)+\kappa^{-2}\lambda}\right) \tag{5.38}$$

For $NU \geq 2$, the expression for $\mathbf{h}$ becomes much more complicated, hence they are not included in this analysis.

5.4.2 Linear GPC polynomial $S(q)$

Using equations-(5.25), (5.31) and (5.33) for the model in equation-(5.32), $S(q)$ is expressed as:

$$S(q) = s_1 + s_2 q^{-1} \tag{5.39}$$

where

$$\begin{aligned} s_1 &= \frac{\sum_{i=d+1}^{N_2} x_i f_{i1}}{\sum_{i=d+1}^{N_2} x_i^2 + \lambda} \qquad\qquad \text{note: } h_i = 0 \text{ for } i \leq d \\ s_2 &= \frac{\sum_{i=d+1}^{N_2} x_i f_{i2}}{\sum_{i=d+1}^{N_2} x_i^2 + \lambda} \end{aligned} \tag{5.40}$$

Substituting equations-(5.18), (5.36) and (5.38) into (5.40) leads to:

$$\begin{aligned} s_1 &= \frac{1}{b}\left\{\frac{(N_2-d)(1-c)-a^2(c-a)\mathcal{P}_2(N_2-d)-a((1-c)-(c-a))\mathcal{P}_1(N_2-d)}{N_2-d+a^2\mathcal{P}_2(N_2-d)-2a\mathcal{P}_1(N_2-d)+\lambda\kappa^{-2}}\right\} \\ s_2 &= -\frac{a}{b}\left\{\frac{(N_2-d)(1-c)-a(c-a)\mathcal{P}_2(N_2-d)-((1-c)a-(c-a))\mathcal{P}_1(N_2-d)}{N_2-d+a^2\mathcal{P}_2(N_2-d)-2a\mathcal{P}_1(N_2-d)+\lambda\kappa^{-2}}\right\} \end{aligned} \tag{5.41}$$

5.4.3 Linear GPC polynomial $T(q)$

From equation-(5.31), $T(q)$ is expressed as:

$$T(q) = C_c(q) + [t_1 q^{-1} + t_2 q^{-2} + \ldots + t_d q^{-d}] \tag{5.42}$$

where from equations-(5.31), (5.38), (5.18) and (5.36) and using Maple, it is shown that:

$$\begin{aligned} t_1 &= \frac{(N_2-d)(1-c)+a^2(a-c)\mathcal{P}_2(N_2-d)-a(1+a-2c)\mathcal{P}_1(N_2-d)-c\lambda\kappa^{-2}}{N_2-d+a^2\mathcal{P}_2(N_2-d)-2a\mathcal{P}_1(N_2-d)+\lambda\kappa^{-2}} \\ t_2 &= \frac{(N_2-d)(1-c)+a^3(a-c)\mathcal{P}_2(N_2-d)-a(1+a^2-c(1+a))\mathcal{P}_1(N_2-d)}{N_2-d+a^2\mathcal{P}_2(N_2-d)-2a\mathcal{P}_1(N_2-d)+\lambda\kappa^{-2}} \end{aligned}$$

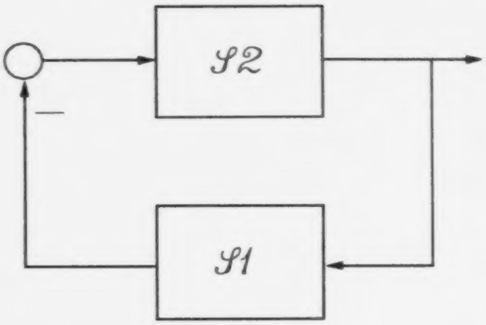


Figure 5.1: $\mathcal{S}_1 - \mathcal{S}_2$ structure for the Small Gain Theorem.

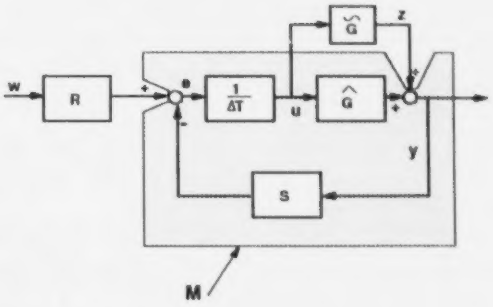


Figure 5.2: Feedback structure of GPC.

$$t_3 = \frac{(N_2 - d)(1 - c) + a^4(a - c)\mathcal{P}_2(N_2 - d) - a(1 + a^3 - c(1 + a^2))\mathcal{P}_1(N_2 - d)}{N_2 - d + a^2\mathcal{P}_2(N_2 - d) - 2a\mathcal{P}_1(N_2 - d) + \lambda\kappa^{-2}}$$

$$\vdots \tag{5.43}$$

5.5 Robust design of GPC

Robust design of a feedback system within the SGT framework involves dividing the closed loop system into two interconnected subsystems $\mathcal{S}_1$ and $\mathcal{S}_2$ as shown in Figure 5.1. Such a system is guaranteed to be stable if $|\mathcal{S}_1\mathcal{S}_2| < 1$ at all the frequencies. If the subsystem $\mathcal{S}_2$ corresponds to the MPM, then for a SISO system, the SGT criterion translates the stability requirement into:

$$|\mathcal{S}_2(\omega)| < \left|\frac{1}{\mathcal{S}_1(\omega)}\right| \qquad \forall \omega \in (0, \omega_N) \tag{5.44}$$

where ω is the frequency span and ω_N is the Nyquist frequency. The criteria given by equation-(5.44) then allows one to examine the stability of GPC in the presence of model-plant mismatch.

Note 5.6 *The frequency form for the discrete domain is given by $e^{-j\omega T_s}$ ($j = \sqrt{-1}$ and T_s is sampling time), however for the sake of notational simplicity, this argument is designated simply by ω in equation-(5.44) and also throughout this chapter.*

The $\mathcal{S}_2$ or MPM in Figure 5.1 or in equation-(5.44) can be represented in several ways, for example, the ***additive***, ***multiplicative*** and ***feedback*** uncertainties [17, 27]. The additive, multiplicative and feedback mismatches are designated by $\tilde{G}(\omega)$, $\tilde{G}_m(\omega)$ and $\tilde{G}_f(\omega)$ respectively and they are expressesd as:

$$\begin{aligned}
\tilde{G}(\omega) &= G(\omega) - \hat{G}(\omega) \\
\tilde{G}_m(\omega) &= \frac{\tilde{G}(\omega)}{\hat{G}(\omega)} \\
\tilde{G}_f(\omega) &= -\frac{\tilde{G}(\omega)}{\hat{G}(\omega)(\hat{G}(\omega) + \tilde{G}(\omega))}
\end{aligned} \tag{5.45}$$

where in the above expressions $G(\omega)$ and $\hat{G}(\omega)$ respectively denotes spectrum of the true plant and its estimated model. The above equations represent the MPMs in frequency domain and are nonparametric so as to include all structural mismatchs such as the uncertainties in model order, time constant, gain, delays or other unusual high frequency dynamics.

$\mathcal{S}_1$ in the $\mathcal{S}_1 - \mathcal{S}_2$ in Figure 5.1 represents the designed closed loop transfer function corresponding to $\mathcal{S}_2$. The expression for $\mathcal{S}_1$ depends on how the MPM *i.e.* $\mathcal{S}_2$ is represented. For example, Figure 5.2 shows linear feedback form of GPC where the plant $G(q)$ is expressed as a summation of the model $\tilde{G}(q)$ and the additive perturbation *i.e.* $\mathcal{S}_2(q) = \tilde{G}(q)$. For this system (in Figure 5.2), the transfer function blocks in the shaded region is lumped into $M(q)$, which is a design variable. $M(q)$ corresponds to $\mathcal{S}_1(q)$ in Figure 5.1 which is given by:

$$M(q) = \frac{S(q)}{\Delta T(q) + \hat{G}(q)S(q)} \tag{5.46}$$

Similarly for $\tilde{G}_m(\omega)$ and $\tilde{G}_f(\omega)$, the corresponding $M_m(\omega)$ and $M_f(\omega)$ are given by:

$$\begin{aligned}
M_m(\omega) &= \frac{S(\omega)\hat{G}(\omega)}{\Delta T(\omega) + S(\omega)\hat{G}(\omega)} \\
M_f(\omega) &= \frac{\hat{G}(\omega)/\Delta T(\omega)}{\Delta T(\omega) + S(\omega)\hat{G}(\omega)}
\end{aligned} \tag{5.47}$$

Thus for the additive, multiplicative and feedback purturbations, the SGT criteria respectively becomes: $|M(\omega)\tilde{G}(\omega)| < 1$, $|M_m(\omega)\tilde{G}_m(\omega)| < 1$ and $|M_f(\omega)\tilde{G}_f(\omega)| < 1$.

Note 5.7 *In equation-(5.47), $M(q)$ and $M_m(q)$ is also known as the mixed sensitivity function and sensitivity function respectively.*

Further analysis presented in this chapter pertains to the additive purturbation because: (a) the additive purturbation is simple to visualize and (b) as shown later, it is often informative to plot $|1/M(\omega)|$ (as in equation-(5.44)) in order to see the frequencies at which stability is most vulnerable. Equations-(5.44), (5.45) and (5.46) are combined to obtain the following robustness criteria for the GPC:

$$\left|\tilde{G}(\omega)\right| < \left|\frac{\Delta T(\omega) + \hat{G}(\omega)S(\omega)}{S(\omega)}\right| \qquad \forall\omega \in [0, \omega_N] \tag{5.48}$$

Some key features of the stability criterion given by equation-(5.48) are:

- This stability criteria can be expressed as a simple graphical tuning procedure for the robust design of GPC.
- Violation of the stability criteria even with exact knowlwdge of uncertainty *may or may not* lead to instability, because SGT provides only a sufficient condition for stability.
- It is more likely that the system will become unstable if the SGT criterion is violated in the mid to higher frequency ranges [28].
- The safest practice would be to design a controller whose $|1/M(\omega)|$ bound exceeds the uncertainty at all frequencies.
- This approach gives a less conservative design compared with $|\tilde{G}(\omega)M(\omega)|_\infty < 1$.
- This method can be use to select tuning parameters for GPC using optimization methods.

Related to the SGT based robustness criterion is the robustness performance specification [27] given by:

$$\| |M(\omega)\tilde{G}(\omega)| + |\hat{S}(\omega)W(\omega)| \|_\infty < 1$$

where $\hat{S}$ in the above equation is the designed sensitivity function and W is the upper bound on $\hat{S}$. As in the case of SGT, the above equation is also applicable in the frequency range $[0, \omega_N]$. However the present study is based only the SGT criteria, as the objective here is to examine the robustness of the GPC tuning parameters.

As mentioned earlier a number of tuning parameters are associated with GPC (and also with other LRPCs). The sheer number of these tuning parameters may appear to be a disadvantage. However, they provide flexibility to handle different problems and to improve computational efficiency. Once the basic tuning parameters such as N_1, N_2, NU and $C_c(q)$ are set, other parameters such as λ, $P(q)$, γ_y, or γ_∞ are adjusted to fine tune GPC. Also the use of optimization methods to select some of the GPC tuning parameters are discussed in Section 5.17.

Clarke and Mohtadi (1989), McIntosh *et al.* (1991) and Mohtadi (1988) have examined the role of these GPC parameters and provided tuning guidelines [20, 26, 29]. The alternative approach for tuning GPC presented here is based on shaping the frequency domain stability margins. This approach also provides a further insight into the role played by these tuning parameters in imparting stability. Another advantage of this approach is that a set of GPC tuning parameters can be estimated *a priori* using simulations that guarantees stability for a known uncertainty bound when implemented on an actual plant.

The following definitions pertaining to the robustness of GPC are used in the subsequent studies.

Definition 5.1 Robustness bound: *The spectrum of* $|1/M(\omega)|$ *corresponding to the additive purturbation is defined as the robustness bound. The robustness bound is sometimes referred as the stability bound because with the violation of the* SGT *criteria, the system can become unstable.*

Definition 5.2 Robustly stable: *The design of GPC is said to be robustly stable if the robustness bound exceeds the spectrum of the mismatch* $|\tilde{G}(\omega)|$ *at all the frequencies.*

Definition 5.3 Robustness margin: *The difference in magnitude (or the gap) between the robustness bound and the* $|\tilde{G}(\omega)|$ *is called the robustness margin.*

Definition 5.4 Robust: *A set of tuning parameters is defined as more robust than a given reference set of tuning parameters, if the robustness bound corresponding to the specified set exceeds the robustness bound of the reference set.*

The stability criteria given by equation-(5.48) leads to the following *lemma* :

Lemma 5.2 *The steady state gain of the robustness bound in equation-(5.48) is equal to the steady state gain of the process model.*

Proof: Equation-(5.46) can be expressed as:

$$\begin{aligned} \left|\frac{1}{M(\omega)}\right| &= \left|\hat{G}(\omega) + \mathcal{C}(\omega)\right| \\ &= \left|\frac{\hat{B}(\omega)}{\hat{A}(\omega)} + \frac{\Delta T(\omega)}{S(\omega)}\right| \end{aligned} \tag{5.49}$$

where $\hat{G}(\omega) = \hat{B}(\omega)/\hat{A}(\omega)$ and $\mathcal{C}(\omega) = \Delta T(\omega)/S(\omega)$ are the expressions for the model and the controller-inverse respectively. Since $\Delta(\omega) = 1 - e^{-j\omega}$, at steady state $(\omega = 0)$, $\Delta(\omega) = 0$, and substitution of this condition in equation-(5.49) proves the *lemma*-5.2.

The following useful corollaries to *lemma*-5.2 can now be stated.

Corollary 5.1 *As* $\omega \to 0$, *the characteristics of robustness bound are influenced more by the process model than by the controller.*

Proof: As $\omega \to 0$, $\Delta(\omega) \to 0$, implying $\Delta T(\omega)/S(\omega) \to 0$; *i.e.* the influence of the controller parameters is reduced relative to the influence of the process model on the robustness bound.

Corollary 5.2 *Stability of SISO GPC cannot be guaranteed if the steady state gain of the uncertainty exceeds the estimated steady state gain of the model.*

Proof: At steady state ($\omega = 0$), the following stability relation can be deduced from equation-(5.49) and *lemma*-5.2:

$$|\tilde{G}(\omega)| < \left|\frac{\hat{B}(\omega)}{\hat{A}(\omega)}\right| \qquad \text{at } \omega = 0 \tag{5.50}$$

If $|\tilde{G}(\omega)|$ is assumed to be a continuous function in the region around $\omega = 0$ (which is true for stable systems) and the robust stability condition is violated at $\omega = 0$ then, through continuity arguments, the stability criterion will also be violated in the region $0 \leq \omega < \omega^{+}$ in the neighbourhood of $\omega = 0$, thus proving the corollary.

Remark 5.1 *The robust stability criteria will not be violated if* $|\tilde{G}(\omega)| < \frac{1}{2}|G(\omega)|$ *at* $\omega = 0$.

Remark 5.2 *Corollary-5.2 is particularly useful for* ω *in the range* $0 \leq \omega < \omega^{+}$. *Since the SGT condition is only sufficient and fairly conservative in the low frequencies [28], it is likely to result in an unstable region for relatively large values of* ω^{+}.

Remark 5.3 *At higher frequencies,* $|\Delta(\omega) \to 1|$, *thus signifying that robustness bound will be more influenced by the controller* $\Delta T(\omega)/S(\omega)$ *at these frequencies. If the magnitude spectrum of the model rolls off rapidly at these higher frequencies, then the robustness bound will be dominated by the controller there (see equation-(5.49).*

It is obvious from equation-(5.49) that the polynomials $\Delta T(\omega)$ and $S(\omega)$ (that constitute the controller inverse $C(q)$), affect the shape of the robustness bound. Also the discussions in the earlier sections (*i.e.* in Sections 5.3, 5.4 and 5.5) showed that it is difficult to obtain generalized expressions for $T(q)$ and $S(q)$ when the model order is greater than one. Therefore it is difficult to ascertain analytically, how the controller inverse $C(q)$ precisely influences the behavior of the robustness bound, when the model order exceeds one. Nevertheless, the robustness of GPC tuning parameters can be ascertained through simulations under all situations. Hence, simulation studies are presented in the ensuing sections that illustrated the robustness of GPC tuning parameters via example dependent case studies. These simulation studies are also supplemented by analytical proofs that apply to first order model under certain assumptions.

5.6 Robustness of GPC tuning parameters

The robustness of GPC parameters is evaluated using a contrived example, because in reality the nature of the true plant and/or the MPM is never known. The plant under consideration is a third order overdamped (true) plant: $G(s) = 1/(s+1)(3s+1)(5s+1)$ whose discrete equivalent for a sample time $T_s = 1$ is:

$$G(q^{-1}) = \frac{0.0077q^{-1} + 0.0212q^{-2} + 0.0036q^{-3}}{1 - 1.19031q^{-1} + 1.1514q^{-2} - 0.2158q^{-3}} \tag{5.51}$$

The above plant (equation-(5.51)) is approximated by the following 1^{st} order model with two numerator parameters (by applying least-squares techniques and by generating using a square wave excitation) as:

$$\hat{G}(q^{-1}) = \frac{0.0419q^{-1} + 0.0719q^{-2}}{1 - 0.8969q^{-1}} \tag{5.52}$$

The mismatch between model and plant is deliberate, because it is desired to examine the robust stability of GPC in the presence of MPM. Noise-free simulations studies are presented for the sake of clarity in understanding the stability properties and the performance of GPC tuning parameters. Conclusions based on noise-free studies are also applied to the system corrupted with noise/disturbances in Section 5.14 and experimentally evaluated in Section 5.15.

A motivational example

The following example highlights the relevance of the work presented in this chapter:
◇ For the plant-model system given by equations-(5.51) and (5.52), suppose the following two sets of GPC tuning parameters are given:

- Set-1: $N_1 = 1$, $N_2 = 5$, $NU = 1$, $\lambda = 0.5$, $\gamma_\infty = 0$, $P(q) = 1$, and $C_c(q) = 1 - 0.5q^{-1}$
- Set-2: $N_1 = 1$, $N_2 = 8$, $NU = 1$, $\lambda = 0.5$, $\gamma_\infty = 0$, $P(q) = 1$, and $C_c(q) = 1 - 0.8q^{-1}$

For these two sets of tuning parameters is it possible to determine the following before performing the closed-loop test:

- If both the above tuning sets will result in stable systems ?
- Which one of the above tuning set is more robust to modelling errors ?

Undoubtly the issues raised by the above questions acquire importance in the context of real-time control applications where the plant dynamics are not fully known.

Besides the issue of robustness, there is also a question of how much robustness margin should be acceptable and in that case one steps into the robustness verses performance

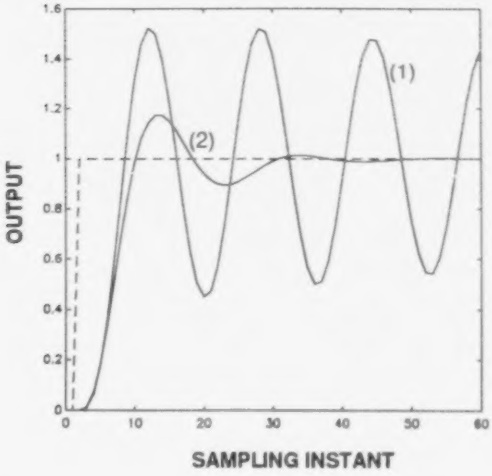


Figure 5.3: Servo responses for tuning sets 1 and 2.

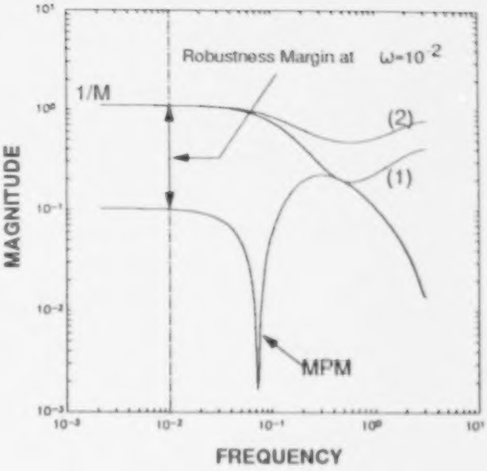


Figure 5.4: Robustness bounds or robustness margins for tuning sets 1 and 2.

issues. It is well known that the robustness is achieved at the cost of controller performance; but a certain degree of robustness to modelling errors is something which cannot be compromised for the sake of stability. The issue of robustness with the perspective of controller performance is pursued further in Chapter 7, whereas the focus of this chapter remains on the robustness properties of the GPC tuning properties.

The closed-loop performance of GPC in Figure 5.3 shows that the response is oscillatory for the tuning set-1 whereas for tuning set-2, the system is stable. For tuning set-1, Figure 5.4 shows that the stability bound touches the uncertainty at a certain frequency thus lacking necessary robustness margin at that frequency, and this gives rise to persistent oscillations in the system response. Whereas for the tuning set-2, sufficient robustness margin exists and as a result, the process response is stable.

Framework for the analytical proof

Equation-(5.49) shows that the robustness bound is influenced by both $\hat{G}(\omega)$ and $C(\omega)$ as shown by the Nyquist plots of $\hat{G}(\omega)$, $C_1(e^{-j\omega})$ and $C_2(e^{-j\omega})$ in Figure 5.5. The controller inverses $C_1(e^{-j\omega})$ and $C_2(e^{-j\omega})$ corresponds to the GPC tuning sets 1 and 2. It is obvious that for a specified $\hat{G}(\omega)$, the changes in the robustness bounds are due to changes in $C(\omega)$ under the influence of different tuning values.

The influence of $C(\omega)$ on $|1/M(\omega)|$ can be analysed by considering the the vectors

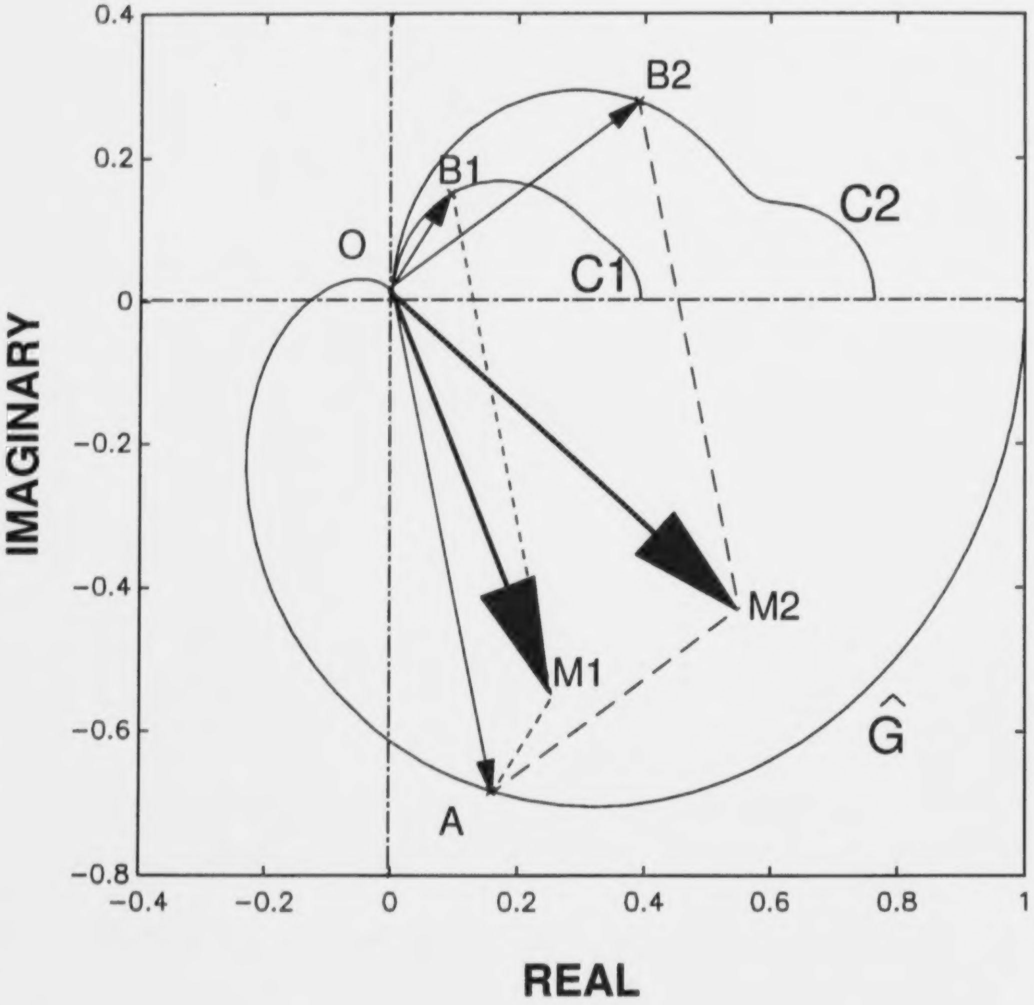


Figure 5.5: Nyquist plots of $\hat{G}(\omega)$, $\mathcal{C}_1(\omega)$ and $\mathcal{C}_2(\omega)$.

$\vec{OA}$, $\vec{OB}_1$ and $\vec{OB}_2$ corresponding to $\hat{G}(\omega)$, $\mathcal{C}_1(\omega)$ and $\mathcal{C}_2(\omega)$ respectively at a given frequency ω as depicted in Figure 5.5. The robustness bounds at ω are then given by $\vec{OM}_1 = \vec{OA}+\vec{OB}_1$ and $\vec{OM}_2 = \vec{OA} + \vec{OB}_2$. At ω the bound $\vec{OM}_2$ is said to be more robust than $\vec{OM}_1$ if $\left|\vec{OM}_2\right| > \left|\vec{OM}_1\right|$. Obviously the robustness is governed by the relative magnitudes and angles of $\vec{OB}_1$ and $\vec{OB}_2$ as $\vec{OA}$ is fixed. The analytical analysis becomes straightforward if it can be shown that $\left|\vec{OB}_2\right| > \left|\vec{OB}_1\right| \Rightarrow \left|\vec{OM}_2\right| > \left|\vec{OM}_1\right|$. However it is nontrivial to show this for $na > 1$ because of the influence of $\angle\vec{OB}_1$[7] and $\angle\vec{OB}_2$.

In other words, since the robustness bound is a complex sum of $\hat{G}(\omega)$ and $\mathcal{C}(\omega)$, it *cannot* be inferred that with the increase in $|\mathcal{C}(\omega)|$, there is also a corresponding increase in magnitude spectrum of the robustness bound $|1/M(\omega)|$. Brute force Algebraic methods are therefore used for a first order model to arrive at the conditions under which it is shown that an increase in $|\mathcal{C}(\omega)|$ causes a corresponding increase in $|1/M(\omega)|$.

In order to analyze the effect of individual tuning parameters on the robustness bound, it is necessary that their effect be examined individually. In the subsequent sections the effects of various GPC tuning parameters on the robustness bounds, and they are complemented by time-domain simulations are shown. For some cases of GPC parameters, the simulation results are backed-up by analytical proofs for a first order model and under the following assumptions:

Assumption 5.1 *The robustness of each tuning parameter is examined by keeping the other tuning parameters constant. The tuning parameters considered for the analytical methods are* N_2, λ, $C_c(q) = 1 - cq^{-1}$ *(for* $0 < c < 1$*) and* NU.

Assumption 5.2 *The default values of the tuning parameters are:* $N_1 = 1$, $\lambda = 0$, $\gamma_\infty = 0$, $P(q) = 1$, $C_c(q) = 1$ *and* $NU = 1$.

Assumption 5.3 *The delay in the model is zero i.e.* $d = 0$.

Assumption 5.4 $G(q)$ *and* $\hat{G}(q)$ *are stable.*

5.7 Effect of N_2

It is known that the system becomes more robust as the costing horizon, N_2, is increased. This can be verified from Figure 5.6, which shows the spectral plots of the robustness bounds for different values of N_2, superimposed on the magnitude spectrum of the uncertainty. The abscissa corresponds to the frequency points, which are normalized by the Nyquist frequency ω_N. The values of N_2 examined are 2, 5, 10 and 20, with the other tuning parameters kept constant at $N_1 = 1$, $NU = 1$, $\lambda = 0$, $C_c(q) = 1 - 0.7q^{-1}$, $\gamma_\infty = 0$ and $P(q) = 1$.

---

[7] where $\angle$ indicates phase contribution.

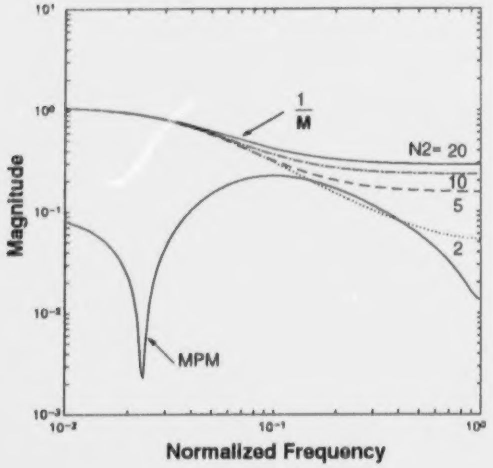


Figure 5.6: Effect of N_2 on the robustness bound.

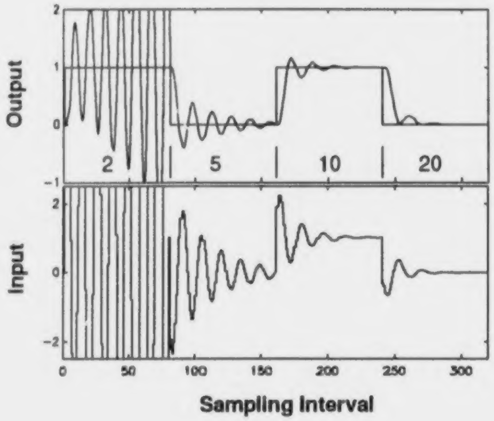


Figure 5.7: Effect of N_2 on the servo tracking.

It is observed that the stability criterion is violated for $N_2 = 2$ giving rise to instability, as can be seen in the servo response in Figure 5.7. The robustness of the controller improves as the value of N_2 increases *i.e.* the robustness margin increases. Also the amount of oscillation in the output reduces with the increase in N_2. For this illustrative example it is therefore required that the value of N_2 be greater than 2. The effect of other tuning parameters that would also give robust control with a smaller values of N_2 are explored in the subsequent sections.

This simulation example shows that robustness margin of the system increases with the increasing value of N_2. Now the question arises: is it possible to analytically establish the conditions under which it can be shown that with the increase in N_2 there is a corresponding increase in the robustness margin? This question is addressed in the following for a first order model.

Lemma 5.3 *Under assumptions 5.1 to 5.4, the magnitude spectrum of $|\mathcal{C}(\omega)|$ increases with an increase in the final prediction horizon N_2.*

Proof: From assumptions 5.1 to 5.4, use of equations (5.31) and (5.33) leads to $T(q) = 1$ which gives:

$$\mathcal{C}(q) = \frac{\Delta}{S(q)} \tag{5.53}$$

therefore the *lemma* reduces to proving that $|S(q)|_{N_2+1} < |S(q)|_{N_2}$. Further, employing equations-(5.25), (5.39) and (5.40) and under stated assumptions, they yield:

$$\begin{aligned} s_1 &= \frac{\sum_{i=1}^{N_2} x_i \sum_{k=0}^{i} a^k}{\sum_{i=1}^{N_2} x_i^2} \\ s_2 &= -\frac{a}{b} \end{aligned} \tag{5.54}$$

which then leads to

$$\frac{s_1|_{N_2+1}}{s_1|_{N_2}} = \left(\frac{\sum_{i=1}^{N_2} x_i^2}{\sum_{i=1}^{N_2} x_i^2 + x_{N_2+1}^2}\right)\left(\frac{x_1(1+a) + \cdots + x_{N_2}\sum_{k=0}^{N_2} a^k + x_{N_2+1}\sum_{k=0}^{N_2+1} a^k}{x_1(1+a) + \cdots + x_{N_2}\sum_{k=0}^{N_2} a^k}\right) \tag{5.55}$$

In order to prove the *lemma*, it needs to be shown that the ratio in equation-(5.55) is less than unity. Rearranging equation-(5.55) and substituting equations-(5.18), (5.36) and (5.34), translates into the following inequality which must hold good for the *lemma* to be true:

$$\left(\frac{1-a^{N_2+2}}{N_2 - (a+a^2)\mathcal{P}_1(N_2) + a^3\mathcal{P}_2(N_2)}\right) < \left(\frac{1-a^{N_2+1}}{N_2 - 2a\mathcal{P}_1(N_2) + a^2\mathcal{P}_2(N_2)}\right) \tag{5.56}$$

The above inequality is always satisfied for $0 < a < 1$ and $N_2 > 1$. Thus the *lemma* is proved because s_2 is constant and the inequality-(5.56) is satisfied.

With the increase in the value of N_2, both the LHS and RHS of equation-(5.56) asymptotically reach zero, indicating that the stability bound reaches an asymptotic value with the increase in the value of N_2; hence the following *corollary* to *lemma*-5.3 can be stated:

Corollary 5.3 *Under the assumptions 5.1 to 5.4, as the value of* N_2 *increases,* $S(\omega)$ *asymptotically reaches:*

$$S(\omega) = \frac{1}{b}(1 - ae^{-j\omega}) \tag{5.57}$$

Proof: From equation-(5.41), the asymptotic value of s_1 becomes:

$$s_1|_{N_2\to\infty} = \frac{1}{b} \tag{5.58}$$

Since s_2 is constant, substituting equations-(5.54) in (5.39) proves the corollary.

Now it is required to investigate the conditions under which it can be shown that an increase in $|\mathcal{C}(\omega)|$ also causes a corresponding increase in $|1/M(\omega)|$. Under the stated assumptions-5.1 to 5.4 this is done by using equations-(5.53) to express $|\mathcal{C}(\omega)|$ as:

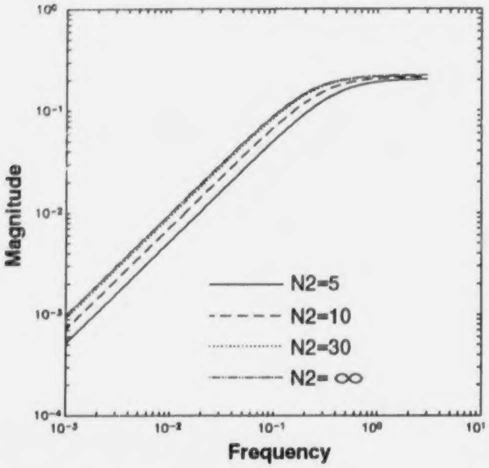


Figure 5.8: Effect of N_2 on the magnitude spectrum of $\mathcal{C}(\omega)$.

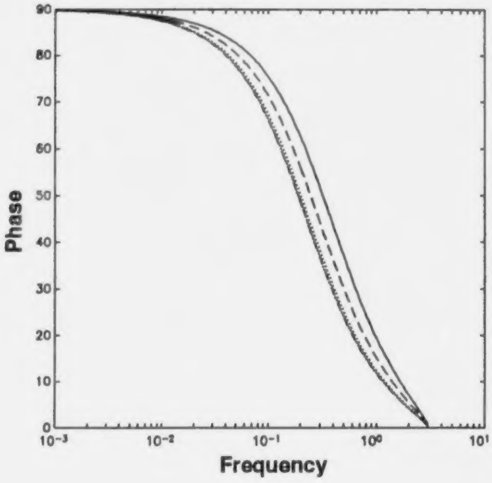


Figure 5.9: Effect of N_2 on the phase spectrum of $\mathcal{C}(\omega)$.

$$\begin{aligned}\mathcal{C}(\omega) &= \frac{1-e^{-j\omega}}{s_1+s_2e^{-j\omega}} \\ &= \frac{(s_1-s_2)(1-\cos(\omega))+j(s_1+s_2)\sin(\omega)}{s_1^2+s_2^2+2s_1s_2\cos(\omega)} \\ &= \left|\sqrt{\frac{2(1-\cos(\omega))}{s_1^2+s_2^2+2s_1s_2\cos(\omega)}}\right| \angle\tan^{-1}\left\{\frac{(s_1+s_2)\sin(\omega)}{(s_1-s_2)(1-\cos(\omega))}\right\}\end{aligned} \tag{5.59}$$

From equations-(5.59) and (5.41) it is evident that $\mathcal{C}(\omega)$ provides a phase lead; at $\omega = 0$, $\mathcal{C}(\omega) = |0|\angle\pi/2$ and at $\omega = \pi$ (*i.e.* at ω_N), $\mathcal{C}(\omega) = |2/\sqrt{s_1^2+s_2^2-2s_1s_2}|\angle\pi$. Similarly the model given by equation-(5.32) can be written as:

$$\begin{aligned}\hat{G}(\omega) &= \frac{be^{-j\omega}}{1-ae^{-j\omega}} \\ &= \left|\frac{b}{\sqrt{1+a^2-2a\cos(\omega)}}\right| \angle\tan^{-1}\left\{\frac{\sin(\omega)}{\cos(\omega)-a}\right\}\end{aligned} \tag{5.60}$$

From equations-(5.60) and (5.59) it is concluded that the relative angle between $\hat{G}(\omega)$ and $\mathcal{C}(\omega)$ varies from $\pi/2$ to π as ω changes from 0 to π (*i.e.* ω_N). Therefore from equations-(5.41) and (5.59), it is deduced that:

$$|\mathcal{C}(\omega)|_{N_2+1} > |\mathcal{C}(\omega)|_{N_2} \qquad \text{and} \qquad \angle\mathcal{C}(\omega)|_{N_2+1} < \angle\mathcal{C}(\omega)|_{N_2} \tag{5.61}$$

which is verified in Figures 5.8 and 5.9 for the system equations-(5.51) and (5.52)[8], which shows that: (a) $|\mathcal{C}(\omega)|$ increases as the frequency and N_2 increase, (b) $\angle\mathcal{C}(\omega)$ reduces with the increases in both frequency and N_2, (c) $|\mathcal{C}(\omega)|$ and $\angle\mathcal{C}(\omega)$ reach an asymptotic limit as $N_2 \to \infty$.

The following terms are defined with reference to Figure 5.5 for notational simplicity:

$$\begin{aligned} &|\vec{OA}| = \rho,\ |\vec{OB}_2| = \sigma,\ |\vec{OB}_1| = \beta\sigma \text{ where } 0 < \beta < 1 \\ &\angle\vec{OB}_2 = \alpha,\ \angle\vec{OB}_1 = \alpha + d\alpha \text{ where } \alpha \text{ is w.r.t. positive real-axis.} \end{aligned} \tag{5.62}$$

From equations-(5.39), (5.41), (5.54) and (5.59) it is shown that:

$$\begin{aligned} \beta(\omega) &= \frac{|\mathcal{C}(\omega)|_{N_2}}{|\mathcal{C}(\omega)|_{N_2+1}} \\ &= \sqrt{\frac{[s_1^2 + s_2^2 + 2s_1s_2\cos(\omega)]|_{N_2+1}}{[s_1^2 + s_2^2 + 2s_1s_2\cos(\omega)]|_{N_2}}} \end{aligned} \tag{5.63}$$

Using Figure 5.5 as a reference and from equation-(5.62) it is clear that $\angle(\vec{OA} + \vec{OB}_2) = \alpha + |\phi|$ and $\angle(\vec{OA} + \vec{OB}_1) = \alpha + d\alpha + |\phi|$ where $\angle\hat{G}(\omega) = \phi$ and $(\alpha + |\phi|, \alpha + d\alpha + |\phi|) > \pi/2$, which then lead to:

$$\begin{aligned} |\vec{OM}_2| &= \sqrt{\rho^2 + \sigma^2 + 2\rho\sigma\cos(\alpha + |\phi|)} \\ |\vec{OM}_1| &= \sqrt{\rho^2 + \beta^2\sigma^2 + 2\beta\rho\sigma\cos(\alpha + d\alpha + |\phi|)} \end{aligned} \tag{5.64}$$

where $|\vec{OM}_2|$ corresponds to $N_2 + 1$ and $|\vec{OM}_1|$ to N_2. Further, $\angle\alpha + |\phi|$ and $\angle\alpha + d\alpha + |\phi|$ are greater than $\pi/2$, therefore it leads to the possibility that $\exists\ |\vec{OB}^\star| = \beta^\star\sigma$ on $|\vec{OB}_1|$ s.t. $|\vec{OB}^\star| < |\vec{OB}_1|$ or $\beta^\star\sigma < \beta\sigma$ and yet $|\vec{OB}^\star + \vec{OA}| = |\vec{OB}_2 + \vec{OA}|$ or $|\beta^\star\sigma + \rho| = |\sigma + \rho|$. This implies that $\beta^\star$ can be obtained by equating the equations in (5.64) and by letting $\beta = \beta^\star$, which yields:

$$\beta^\star = \frac{1}{\sigma}\left[\rho\sin(\alpha + d\alpha + |\phi| - \frac{\pi}{2}) \pm \sqrt{\sigma^2 + \rho^2\cos^2(\alpha + d\alpha + |\phi|) + 2\rho\sigma\cos(\alpha + |\phi|)}\right] \tag{5.65}$$

Only one of the roots of equation-(5.65) is meaningful and they are:

---

[8] Although equation-(5.61) is derived for the model-(5.32), Figures 5.8 and 5.9 show that these conditions (*i.e.* equation-(5.61)) also apply to the model-(5.52) that has two parameters in the numerator.

1. $\beta^\star = 1$: This implies $\beta < \beta^\star$ *i.e.* $|\vec{OM}_1| > |\vec{OM}_2|$ eventhough $|\vec{OB}_1| < |\vec{OB}_2|$. This situation is undesirable because with the increase in $|\mathcal{C}(\omega)|$ there is a decrease in $|1/M(\omega)|$.
2. $0 < \beta^\star < 1$: Under this case two possibilities exist *i.e.* $\beta^\star < \beta$ and $\beta < \beta^\star$. In the former case $|\vec{OB}_1| < |\vec{OB}_2| \Rightarrow |\vec{OM}_1| < |\vec{OM}_2|$, which is desirable. The later case (*i.e.* $\beta < \beta^\star$) however is undesirable.
3. $\beta^\star < 0$: This situation is always desirable because in such a case there is always an increase in $|1/M(\omega)|$ with the increase in $|\mathcal{C}(\omega)|$.

$\beta^\star$ is applicable for $\omega > 0$, because at $\omega = 0$, $\sigma = 0$, which makes the solution (5.65) indeterminate. However at the Nyquist frequency (ω_N), $\angle\alpha + d\alpha + |\phi| = \angle\alpha + |\phi| = \pi$, therefore substitution of this condition in equation-(5.65) leads to $\beta^\star(\omega_N) = (2\rho/\sigma) - 1$. Now consider the application of this analysis on the example problem given by equations-(5.51) and (5.52). Figure 5.10 shows that at all ω, $\beta^\star < \beta$ and $\beta^\star(\omega)$ is a decreasing function of ω (since σ increases and ρ decreases with the increase in ω (5.65)) and from Figure 5.5 it is apparant that $\beta(\omega)$ is an increasing function of ω, and also as $\omega \to 0 \Rightarrow \beta^\star(\omega) \to \beta(\omega)$, therefore according to the condition number 2 above, it can be concluded that increasing the value of N_2 also increases the robustness margin as verified in Figure 5.4.

5.8 Effect of λ

The effect of the control weighting term λ is to detune the controller performance of an open-loop stable plant and thus its use gives a sluggish closed-loop response. Hence it is recommended that λ be kept as small as possible. Nevertheless, increasing the value of λ adds to the robustness by moving the robustness bounds further away from the uncertainty, as can be seen from Figures 5.11 and 5.12 for the system given by equations-(5.51) and (5.52). Several λ values are examined, with the other tuning parameters kept constant at $N_1 = 1$, $N_2 = 5$, $NU = 1$, $\gamma_\infty = 0$, $C_c(q) = 1$ and $P(q) = 1$. It is also observed that λ produces a slight 'minimum' in the robustness bounds, thus making the controller sensitive at the frequencies where the dip occurs. For this problem it is seen that the system becomes unstable for $\lambda < 1$, which is confirmed by the plots in Figure 5.11 and 5.12. For $\lambda = 1$, the robustness bounds come very close to the uncertainty, which give a lightly damped oscillatory response for this system. Although increasing λ increases the robustness margin, but for this example, its use does not seem to reduce the amount of oscillations in a signifinant way. Whereas in Figures 5.6 and 5.7 it was seen that the amount of oscillations reduced significantly as the value of N_2 was increased. The effect of λ on the robustness margin for a first order model is explained by the following *lemma*:

Lemma 5.4 *Under assumptions-5.1 to 5.4, $|\mathcal{C}(\omega)|$ increases with the increase in the controller weighting term λ.*

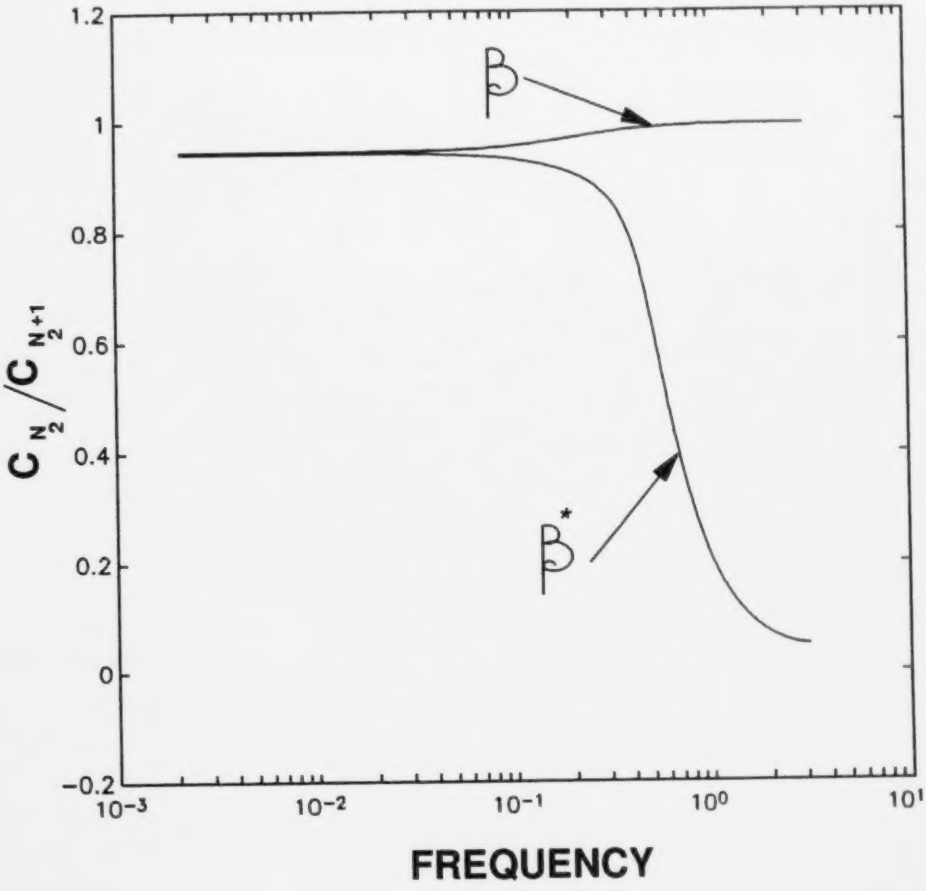


Figure 5.10: Plot of $\beta = |\mathcal{C}(\omega)|_{N_2}/|\mathcal{C}(\omega)|_{N_2+1}$ and $\beta^* = |\mathcal{C}^*(\omega)|_{N_2}/|\mathcal{C}(\omega)|_{N_2+1}$.

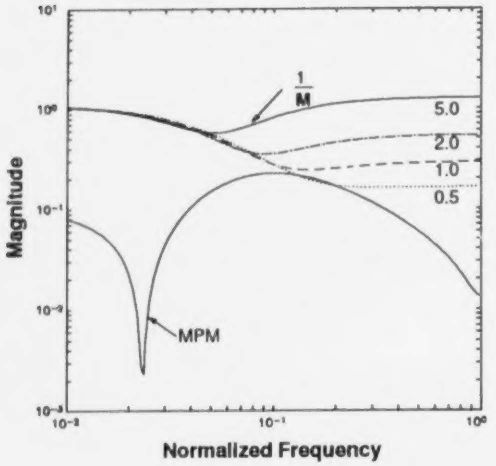


Figure 5.11: Effect of λ on the robustness bound.

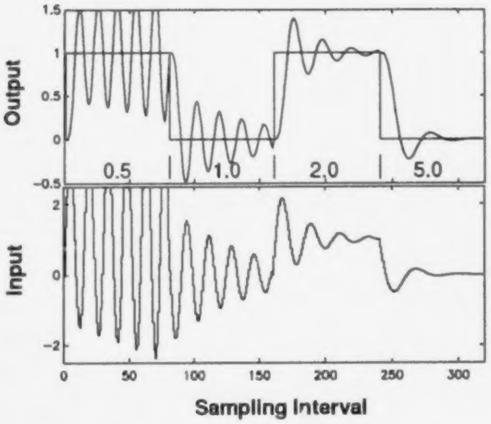


Figure 5.12: Effect of λ on the servo tracking.

Proof: As in the case of *lemma*-5.3, here also it is shown that $T(q) = 1$. Therefore for the *lemma* to be true, it needs to be proved that $|S(\omega)|_{\lambda_2} < |S(\omega)|_{\lambda_1}$ where $\lambda_2 > \lambda_1$. Under the above assumptions and from equation-(5.41), s_1 and s_2 reduces to:

$$s_1|_\lambda = \frac{1}{b}\left(\frac{N_2 + a^3\mathcal{P}_2(N_2) - (a+a^2)\mathcal{P}_1(N_2)}{N_2 + a^2\mathcal{P}_2(N_2) - 2a\mathcal{P}_1(N_2) + \lambda\kappa^{-2}}\right)$$
$$s_2|_\lambda = -\frac{a}{b}\left(\frac{N_2 + a^2\mathcal{P}_2(N_2) - 2a\mathcal{P}_1(N_2)}{N_2 + a^2\mathcal{P}_2(N_2) - 2a\mathcal{P}_1(N_2) + \lambda\kappa^{-2}}\right) \tag{5.66}$$

From the above equation-(5.66) it is clear that:

$$s_i|_{\lambda_2} < s_i|_{\lambda_1} \text{ where } \lambda_2 > \lambda_1 \text{ for } i = 1, 2 \tag{5.67}$$

which implies: $|S(\omega)|_{\lambda_2} < |S(\omega)|_{\lambda_1}$ if $\lambda_2 > \lambda_1$, thus proving the *lemma*.

Remark 5.4 *The values of both s_1 and s_2 changes with the change in λ, whereas N_2 affects only s_1. Therefore the robustness margin is influenced more by λ than by N_2.*

Corollary 5.4 *For a given λ, as $N_2 \to \infty$, $S(\omega)$ asymptotically reaches: $S(\omega)|_{\lambda, N_2\to\infty} = (1 - ae^{-j\omega})/b$.*

Proof: : Equation-(5.66) can be used to show that as $N_2 \to \infty$, $s_1|_{\lambda,N_2\to\infty} = 1/b$ and $s_2|_{\lambda,N_2\to\infty} = -a/b$ thus proving the *corollary*.

Corollary 5.5 *For a given N_2, $S(\omega)$ asymptotically reaches zero as $\lambda \to \infty$.*

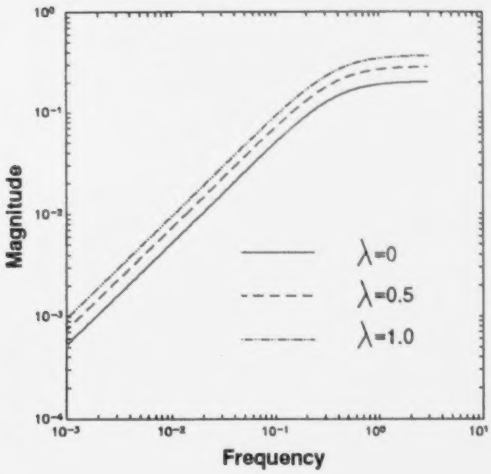


Figure 5.13: Effect of λ on the magnitude spectrum of $\mathcal{C}(\omega)$.

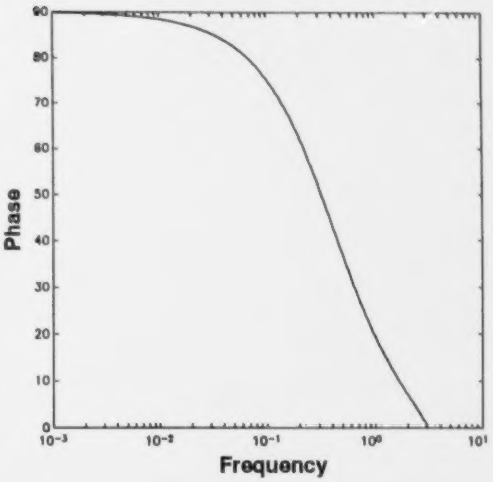


Figure 5.14: Effect of λ on the phase of $\mathcal{C}(\omega)$.

Proof: From equation-(5.66) it is obvious that $s_1 = s_2 = 0$ as $\lambda \to \infty$, thus proving the *corollary.*

The *corollaries*-5.4 and 5.5 are interesting because they convey the following:

- As the value of N_2 increases, the robustness margin reaches an asymptotic limit. This also means that beyond a certain value of N_2, there is no appreciable improvement in the robustness margin.
- The robustness margin increases with increase in the value of λ. At infinite λ, the robustness margin is infinite since for such a case there is no controller action and the process is open loop stable by assumption.

It is now required to establish the conditions under which it can be shown that that with the increase in $|\mathcal{C}(\omega)|$ due to increasing value of λ, there is also an increase in $|1/M(\omega)|$; which is shown in the following:

$$\begin{aligned} \{|O\vec{B}_2| = |\mathcal{C}(\omega)|_{\lambda_2} = \sigma\} &> \{|O\vec{B}_1| = |\mathcal{C}(\omega)|_{\lambda_1} = \beta\sigma\} \\ \{\angle O\vec{B}_2 = \angle \mathcal{C}(\omega)|_{\lambda}\} &= \{\angle O\vec{B}_1 = \angle \mathcal{C}(\omega)|_{\lambda_1}\} \end{aligned} \tag{5.68}$$

The above equation-(5.68) is easily verified using equations-(5.59) and (5.66).

Remark 5.5 *From equation-(5.68) it is clear that the phase of $\mathcal{C}(\omega)$ is not influenced by λ.*

The above remark is verified in Figures 5.13 and 5.14 which shows: (a) as in the case of N_2, $|\mathcal{C}(\omega)|$ increases with the increase in λ and frequency, but (b) λ has no effect on $\angle\mathcal{C}(\omega)$. Now it remains is to be shown that:

$$|\vec{OM_2}|_{\lambda_2} > |\vec{OM_1}|_{\lambda_1} \text{ where } \lambda_2 > \lambda_1 \tag{5.69}$$

From equation-(5.68) it is clear that:

$$\alpha + |\phi| > \frac{\pi}{2} \tag{5.70}$$

which leads to the possibility that $\exists\, |\vec{OB^{\star}}| = \beta^{\star}\sigma$ such that $|\vec{OB^{\star}}| < |\vec{OB_1}|$ or $\beta^{\star}\sigma < \beta\sigma$ and yet $|\vec{OB^{\star}} + \vec{OA}| = |\vec{OB_2} + \vec{OA}|$ or $|\beta^{\star}\sigma + \rho| < |\sigma + \rho|$. Substituting $d\alpha = 0$ in equation-(5.65), $\beta^{\star}$ is obtained as:

$$\beta^{\star} = \left\{\frac{1}{\sigma}\left(2\rho\sin\left(\alpha + |\phi| - \frac{\pi}{2}\right) - \sigma\right), 1\right\} \tag{5.71}$$

At the lower frequencies $\beta^{\star} = 1$ and it drops to $2\sigma/\rho - 1$ at ω_N. Moreover in this case β is constant because from equations-(5.53), (5.59) and (5.66) it is obvious that:

$$\beta = \frac{N_2 + a^2\mathcal{P}_2(N_2) - 2a\mathcal{P}_1(N_2) + \lambda_1\kappa^{-2}}{N_2 + a^2\mathcal{P}_2(N_2) - 2a\mathcal{P}_1(N_2) + \lambda_2\kappa^{-2}} \tag{5.72}$$

Figure 5.15 shows that at the lower frequencies $\beta < \beta^{\star}$ and at the higher frequencies $\beta > \beta^{\star}$, implying that λ imparts robustness at the higher frequencies but not neccessarily at the lower frequencies.

5.9 Effect of $C_c(q)$

One of the most important GPC tuning parameter is the monic observer filter polynomial $C_c(q)$, which helps in rejecting disturbances, unmodelled dynamics and other high frequency noise [26]. The filter thus adds to the robustness in the system. The values of $C_c(q)$ are examined by setting the other tuning knobs constant at $N_1 = 1$, $N_2 = 5$, $NU = 1$, $\lambda = 0$, $\gamma_{\infty} = 0$ and $P(q) = 1$. With these tuning values and for $C_c(q) = 1$, the stability criteria gets violated, giving rise to an unstable system, as shown in Figures 5.16 and 5.17. The robustness bound moves up from the uncertainty for $C_c(q) = 1 - 0.8q^{-1}$, and as a result the system becomes robust to model uncertainties. Figures 5.16 and 5.17 shows that the use of $C_c(q)$ filter also reduces the amount of oscillations in the system response. For $C_c(q) = (1 - 0.8q^{-1})^2$, the robustness margin increases further to give a more robust system although it shows a minimum. The robustness due to the $C_c(q)$ filter is next explored analytically for a first order model.

Lemma 5.5 *Under the assumptions-5.1 to 5.4, the use of $C_c(q)$ filter increases the magnitude spectrum of $\mathcal{C}(\omega)$.*

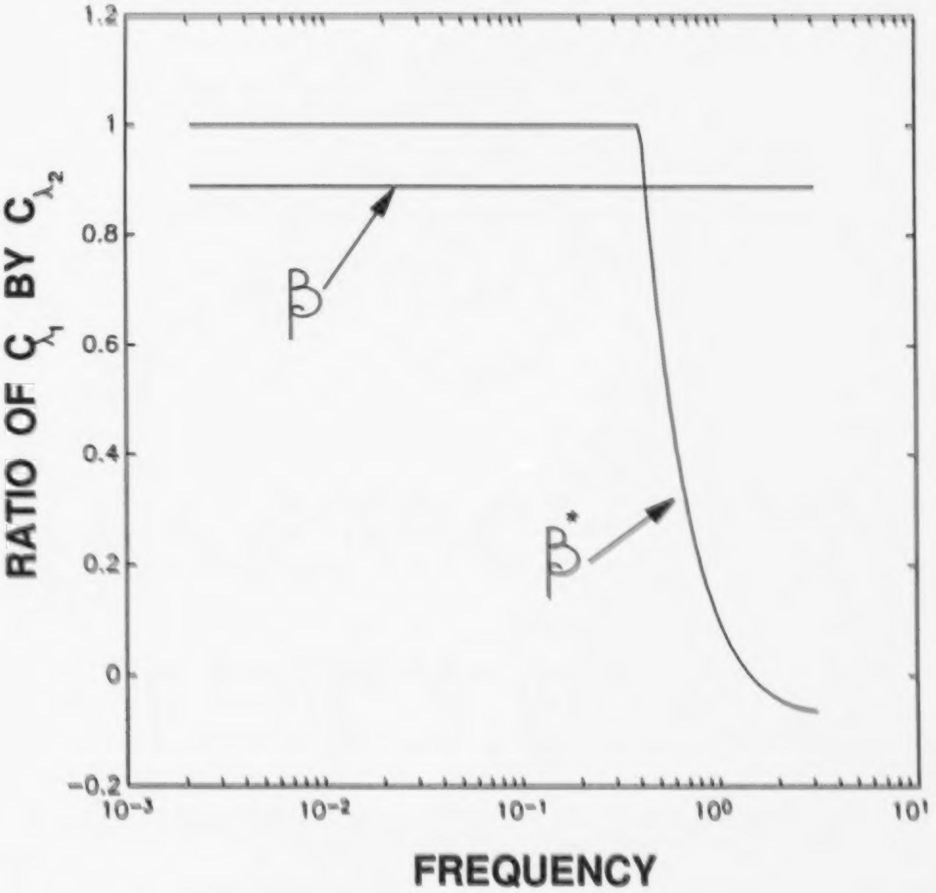


Figure 5.15: Plot of $\beta = |\mathcal{C}(\omega)|_{\lambda_1}/|\mathcal{C}(\omega)|_{\lambda_2}$ and $\beta^* = |\mathcal{C}^*(\omega)|_{\lambda_1}/|\mathcal{C}(\omega)|_{\lambda_2}$

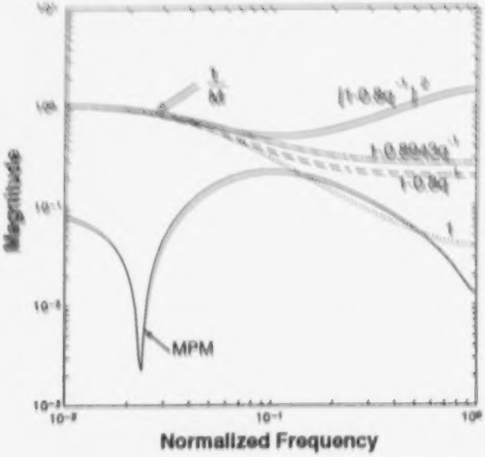


Figure 5.16: Effect of $C_c(q)$ on the robustness bound.

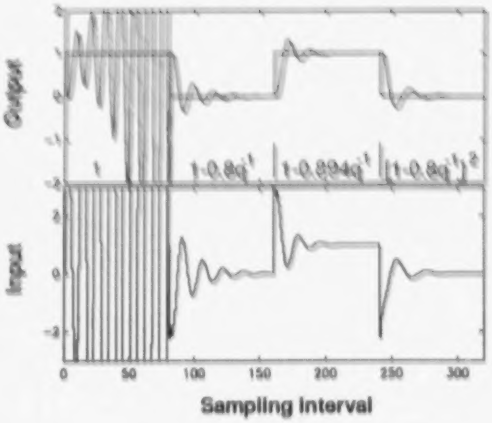


Figure 5.17: Effect of $C_c(q)$ on the servo tracking.

Proof: Under the stated assumptions and for $C_c(q) = 1 - cq^{-1}$, where $0 \leq c < 1$, equations-(5.41) reduces to:

$$
\begin{aligned}
s_1|_{C_c} &= \frac{1}{b}\left(\frac{\mathcal{N}_2(1-c) + (a^3 - ca^2)\mathcal{P}_2(\mathcal{N}_2) - (a + a^2 - 2ca)\mathcal{P}_1(\mathcal{N}_2)}{\mathcal{N}_2 + a^2\mathcal{P}_2(\mathcal{N}_2) - 2a\mathcal{P}_1(\mathcal{N}_2) + \lambda\kappa^{-2}}\right) \\
&= \frac{1}{b}\left(\frac{\mathcal{N}_1}{\mathcal{D}}\right) \\
s_2|_{C_c} &= -\frac{a}{b}\left(\frac{\mathcal{N}_2(1-c) + (a^2 - ac)\mathcal{P}_2(\mathcal{N}_2) - (2a - c(1+a))\mathcal{P}_1(\mathcal{N}_2)}{\mathcal{N}_2 + a^2\mathcal{P}_2(\mathcal{N}_2) - 2a\mathcal{P}_1(\mathcal{N}_2) + \lambda\kappa^{-2}}\right) \\
&= -\frac{a}{b}\left(\frac{\mathcal{N}_2}{\mathcal{D}}\right) \qquad (5.73)
\end{aligned}
$$

Comparing equations-(5.66) and (5.73) it is concluded that:

$$
\begin{aligned}
s_1|_{C_c(q)} &< s_1|_{C_c(q)=1} \\
s_2|_{C_c(q)} &< s_2|_{C_c(q)=1} \qquad (5.74)
\end{aligned}
$$

for $\lambda = 0$ and from equation-(5.74), it is concluded that this *lemma* is true. However when $\lambda > 0$ and in the presence of $C_c(q)$, the polynomial $T(q) = 1 - tq^{-1}$ is also involved. Under the stated assumptions and from equations-(5.31), (5.34) and (5.30), t is simplifies to:

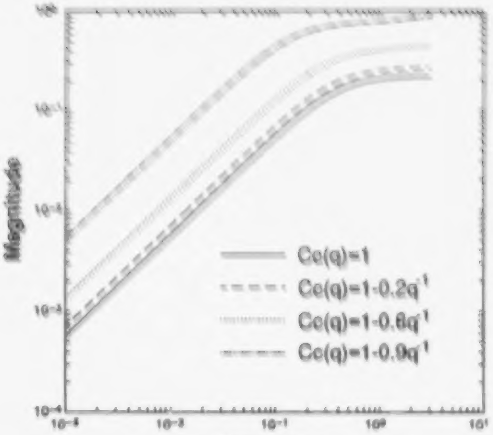


Figure 5.18: Effect of $C_c(q)$ on the magnitude spectrum of $\mathcal{C}(\omega)$.

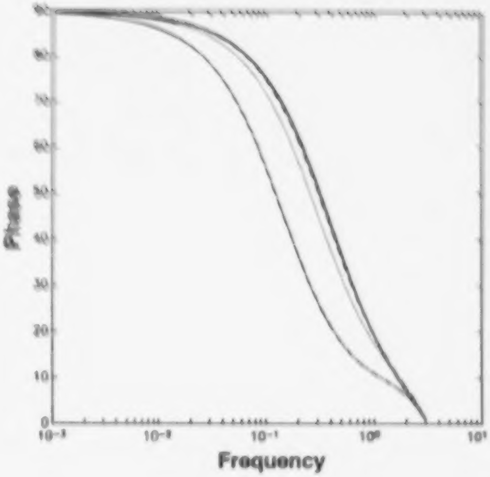


Figure 5.19: Effect of $C_c(q)$ on the phase spectrum of $\mathcal{C}(\omega)$.

$$\begin{aligned} t &= \frac{c\lambda\kappa^{-2}}{\mathcal{N}_2 + a^2\mathcal{P}_2(\mathcal{N}_2) - 2a\mathcal{P}_1(\mathcal{N}_2) + \lambda\kappa^{-2}} \\ &= \frac{\mathcal{N}_t}{\mathcal{D}} \end{aligned} \tag{5.75}$$

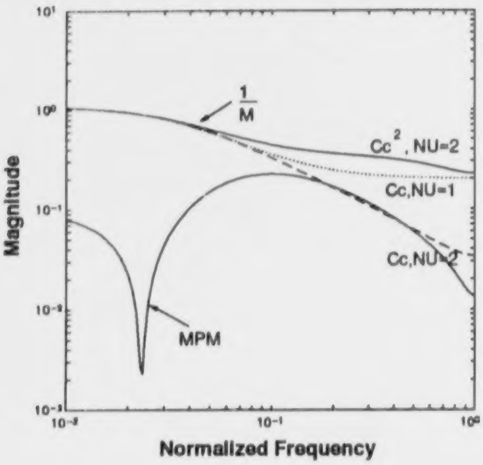
Therefore from equations-(5.73) and (5.75), $|\mathcal{C}(q)| = |\Delta T(q)/S(q)|$ can be expressed as:

$$\begin{aligned} \left|\frac{T\Delta(\omega)}{S(\omega)}\right| &= \sqrt{\frac{2[1-\cos(\omega)][1+t^2+2t\cos(\omega)]}{s_1^2+s_2^2+2s_1s_2\cos(\omega)}} \\ &= \sqrt{\frac{2[1-\cos(\omega)]b^2[\mathcal{D}^2+\mathcal{N}_t^2+2\mathcal{N}_t\mathcal{D}\cos(\omega)]}{\mathcal{N}_1^2+a^2\mathcal{N}_2^2-2a\mathcal{N}_1\mathcal{N}_2\cos(\omega)}} \end{aligned} \tag{5.76}$$

From equations-(5.73), (5.74) and (5.75) it is clear that:

- With an increase in λ, there is increase in $\mathcal{D}$ and $\mathcal{N}_t$ which proves the *lemma* for a fixed case of $C_c(q)$.
- With an increase in c in $C_c(q) = 1 - cq^{-1}$ such that $0 \le c < 1$, there is an increase in $\mathcal{N}_t$ and decrease in $\mathcal{N}_1$ and $\mathcal{N}_2$, which proves the *lemma* for this case.

The phase contribution due to $C_c(q)$ on $\mathcal{C}(\omega)$ is given by:



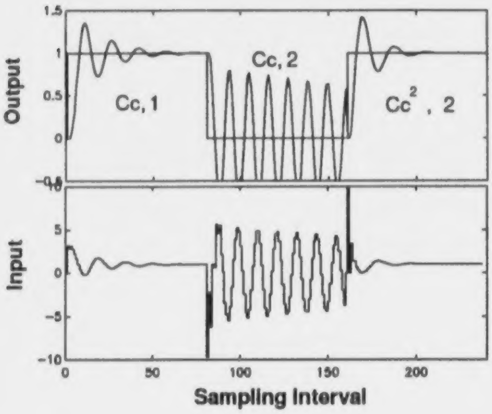


Figure 5.20: Effect of NU on the robustness bound.

Figure 5.21: Effect of NU on the servo tracking.

$$\angle\mathcal{C}(\omega) = \tan^{-1}\left\{\frac{\sin(\omega)\left\{2t\cos(\omega) - (1+t)\right\}}{(1-t) + (1+t)\cos(\omega) - 2t\cos^2(\omega)}\right\} \tag{5.77}$$

The effect of $C_c(q)$ on the magnitude and phase spectrum of $\mathcal{C}(\omega)$ are illustrated in Figures 5.18 and 5.19, which shows that: (a) as in the case of N_2 or λ, $|\mathcal{C}(\omega)|$ increases with the increase in frequency and $C_c(q)$ and (b) as in the case of N_2, $\angle\mathcal{C}(\omega)$ decreases with the increase in frequency and $C_c(q)$. Since the influence of $C_c(q)$ on $\mathcal{C}(\omega)$ is similar to N_2, therefore the analysis carried for N_2 also applies in the case of $C_c(q)$.

5.10 Effect of *NU*

The Control horizon NU adds to the aggressiveness in the control action. It is generally adequate to use $NU = 1$ for open loop stable systems (*e.g.* equations-(5.51) and (5.52)). For the control settings at $N_1 = 1$, $N_2 = 5$, $\lambda = 0$, $\gamma_\infty = 0$, $C_c(q) = 1 - 0.8q^{-1}$ and $P(q) = 1$, a stable response is obtained for $NU = 1$, whereas for NU=2, the stability bound just touches the uncertainty line, giving rise to an oscillatory response with large control actions, as depicted in Figures 5.20 and 5.21. The figure also shows that the oscillations due to $NU = 2$ are attenuated by selecting a second order filter $C_c(q) = (1 - 0.8q^{-1})^2$. The second order filter however cannot prevent the large initial control moves as a result of using $NU = 2$. The effect of NU on the robustness bound is next analytically examined in the following:

Lemma 5.6 *Under the assumptions-5.1 to 5.4, the magnitude spectrum of $C(\omega)$ reduces with the increase in the value of the NU.*

Proof: The general expression for $\mathbf{h}$ becomes complicated for $NU \geq 2$ even for a first order model. Therefore this *lemma* is verified by restricting $N_2 = 2$. Under the assumptions-5.1 to 5.4 and for $N_2 = 2$, $\mathbf{h}$ can be expressed as:

$$\mathbf{h}|_{NU=1,N_2=2} = \left(\frac{1}{b(1+(1+a)^2)}\right)[1 \qquad 1+a] \tag{5.78}$$

similarly for $NU = 2$ and $N_2 = 2$, the first row of $\mathbf{h}$ is shown to be:

$$\mathbf{h}|_{NU=2,N_2=2} = \frac{1}{b}[1 \qquad 0] \tag{5.79}$$

From equations-(5.25), (5.31), (5.78) and (5.79) it follows that:

$$\begin{aligned} s_1|_{N_2=2,NU=1} &= \frac{1+a}{b}\left(1 - \frac{a}{1+(1+a)^2}\right) \\ s_1|_{N_2=2,NU=2} &= \frac{1+a}{b} \end{aligned} \tag{5.80}$$

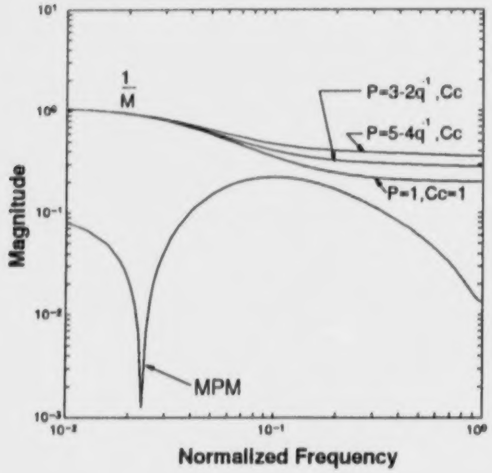
which verifies the *lemma* because s_2 is constant.

5.11 Effect of $P(q)$

The output weighting polynomial $P(q)$ provides model-following characteristics for the GPC. Generally, a detuned model-following strategy is employed to circumvent the problem of zero cancellation [21]. The use of $P(q)$ can give an offset if $P(1) \neq 1$. In Figures 5.22 and 5.23 it is observed that the use of $P(q)$ enhances the robustness by pushing the stability bound up and away from the uncertainty spectrum. Robustness in the system is improved with the increase in the values of $P(q)$. The $P(q)$ polynomials considered are $P(q) = 5 - 4q^{-1}$ and $P(q) = 3 - 2q^{-1}$, with the other tuning parameters at $N_1 = 1$, $N_2 = 5$, $NU = 1$, $\lambda = 0$, $\gamma_\infty = 0$ and $C_c(q) = 1 - 0.8q^{-1}$.

5.12 Effect of γ-weighting

The steady state weighting γ_∞ gives the effect of a large prediction horizon, without actually using a large value for N_2. Therefore as in the case of N_2 or λ, the use of γ_∞ increases the robustness margin as shown in Figure 5.24 for the γ_∞ values of 0, 0.5 and 1.0. The corresponding servo responses for these γ_∞ values shown in Figure 5.25 are obtained by setting other tuning parameters constant at $N_1 = 1$, $N_2 = 5$, $NU = 1$, $\lambda = 0$ $C_c(q) = 1 - 0.7q^{-1}$ and $P(q) = 1$. As in the case of N_2, the servo responses in Figure 5.25 show that the amount of inital oscillations reduces with the increase in γ_∞ values. It has also



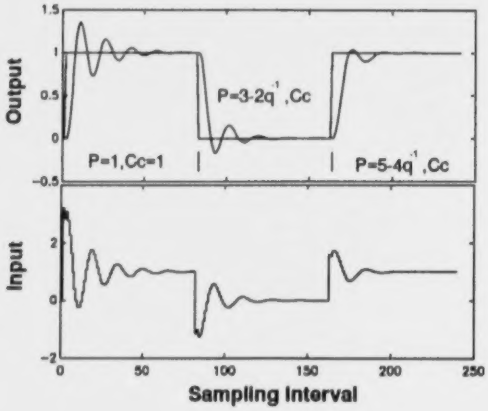
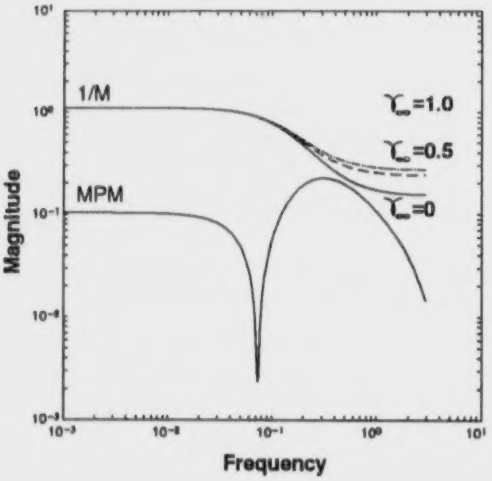


Figure 5.22: Effect of $P(q)$ on the robustness bound.

Figure 5.23: Effect of $P(q)$ on the servo tracking.



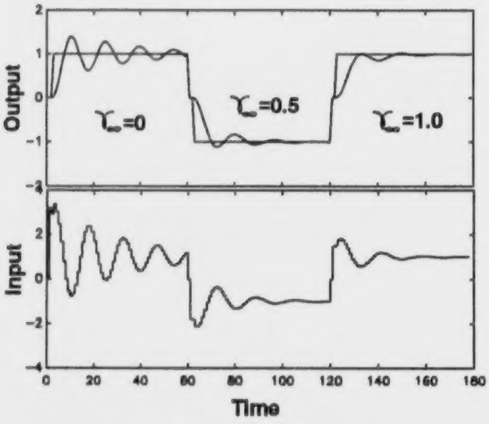


Figure 5.24: Effect of γ_{inf} on the robustness bound.

Figure 5.25: Effect of γ_{inf} on the servo tracking.

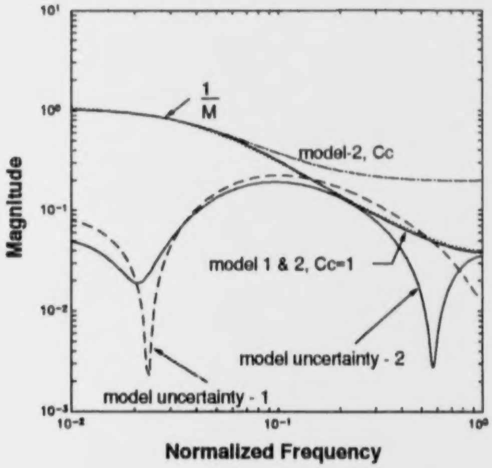


Figure 5.26: Effect of model on the robustness bound.

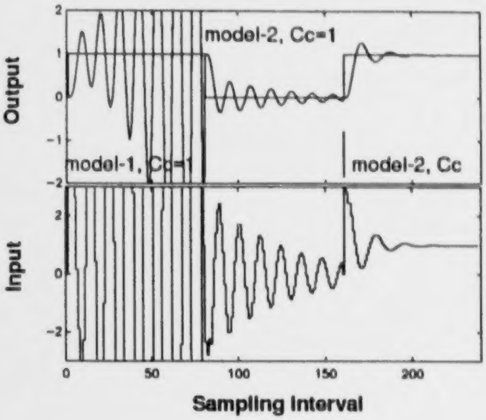


Figure 5.27: Effect of model on the servo tracking.

been shown that the use of γ_∞ improves the stability margin of a system, hence its use is recommended in place of λ [22].

5.13 Effect of model

The impact of model accuracy on the stability and performance of any model-based controller is often profound, and GPC is no exception to this fact. The uncertainty shown by the dashed curve in Figure 5.26 corresponds to the model in equation-(5.52), where the tuning constants $N_1 = 1$, $N_2 = 5$, $NU = 1$, $\lambda = 0$, $\gamma_\infty = 0$, $P(q) = 1$ and $C_c(q) = 1$ result in an unstable system, as shown earlier. If an extra parameter is added to the numerator in the model, then the following model is identified by using least squares fitting:

$$\hat{G}_2(q) = \frac{0.0405q^{-1} + 0.0313q^{-2} + 0.0523q^{-3}}{1 - 0.8836q^{-1}} \tag{5.81}$$

The extra numerator parameter in the new model enables it to capture more dynamics and as a consequence it gives a robust closed-loop system even with the same tuning parameters that resulted in an unstable system for the first model (equation-(5.52)). This highlights the need for proper model validation [30] prior to control. The use of the $C_c(q) = 1 - 0.8q^{-1}$ filter improves the performance for the second model in comparison to the first model, as illustrated in Figure 5.27.

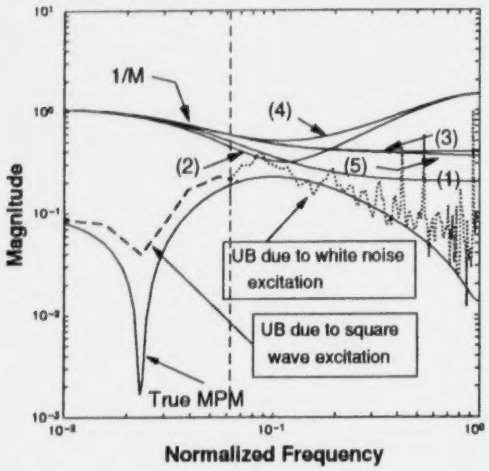


Figure 5.28: Effect of tuning parameters on the robustness bound in the presence of measurement noise and disturbances.

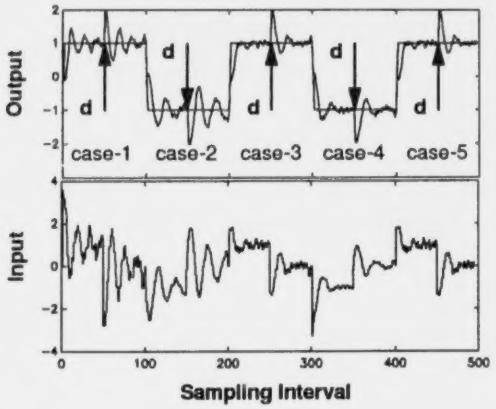


Figure 5.29: Effect of parameters on the servo tracking in the presence of measurement noise and disturbances.

5.14 Robustness of GPC in presence of disturbances

The effect of additive noise in the robust design of GPC is examined in this section for the plant-model system given by equations-(5.51) and (5.52) by adding measurement noise (SNR$\approx$ 3.1) to the process output. The process is also subjected to step type disturbances having the same magnitude and direction as the set point. Robustness bounds and corresponding time domain performance for different cases of tuning parameters listed in Table 5.14 are shown in Figures 5.28 and 5.29. The direction and location of the step type disturbances are symbolically shown in Figure 5.29 by *'d'*. Figure 5.28 also shows the estimated spectral uncertainty from noisy data and the actual MPM. A white noise excitation is used to determine the upper bound on the MPM for higher frequencies. A square wave input is used to find the upper bound on MPM at the lower frequencies. However, in this case accuracy at the lower frequencies is not important because there is sufficient robustness margin at these frequencies.

The stability of a system is affected only when the robustness bound violates the stability condition with respect to the true MPM. That is because a stable characteristic polynomial $1 + G_p(q)G_c(q)$, (where $G_p(q)$ and $G_c(q)$ are true plant and controller respectively) is also stable with respect to measurement noise. The effect of different tuning parameter sets on the performance of the closed-loop system in the presence of measurement noise and disturbances are evaluated for the following cases:

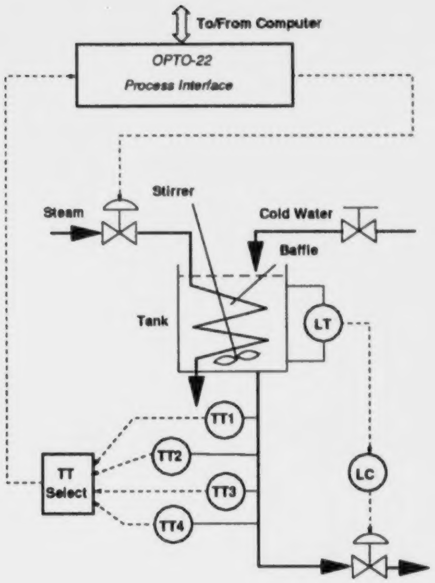


Figure 5.30: Experimental set-up.

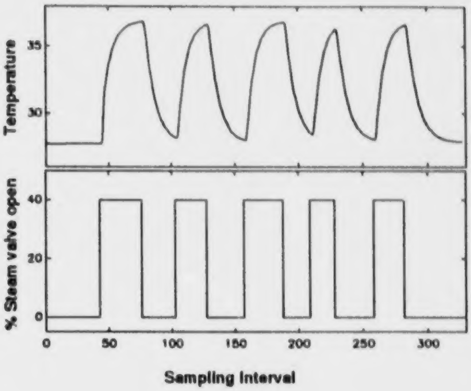


Figure 5.31: Plant input-output data.

- case-1, which shows a significant sensitivity to noise and disturbances, since the robustness bound is closer to the MPM;
- case-2, where an increase in λ causes a minima in the robustness bound, thus bringing $|1/M(\omega)|$ closer to $|\tilde{G}(\omega)|$ and consequently not leading to any significant improvement in performance;
- case-3, where for higher values of N_2 the robustness bound moves further away from $\tilde{G}(\omega)$, and as a result the controller action is detuned and also provides better disturbance rejection;
- case-4, where with a second order $C_c(q)$ filter, the controller gives a better disturbance rejection even for a smaller value of N_2;
- case-5, where the advantage of using $P(q)$ is illustrated, with the spectrum of $|1/M(\omega)|$ almost coinciding with that of case-3; therefore, although the value of N_2 is smaller in this case, its performance is as good as for case-3.

5.15 Experimental evaluation

The signal processing methods for estimating the uncertainty, discussed in Chapter 2, are combined with the GPC robust design techniques presented in the earlier sections of this chapter, to experimentally evaluate the robustness objective of GPC tuning parameters.

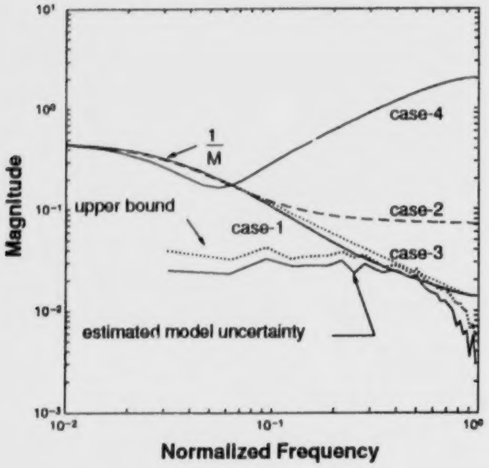


Figure 5.32: Effect of tuning parameters on the robustness bound for the experimental set-up.

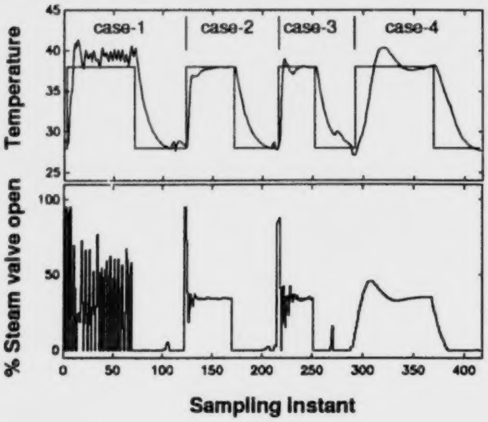


Figure 5.33: Effect of tuning parameters on the servo tracking for the experimental set-up.

A schematic of the experimental set up is shown in Figure 5.30. The experimental plant is a stirred-tank heater in which cold water is heated by a submerged steam coil in an insulated vessel. The plant is open loop stable and operates as a continuous process. Through an OPTOMUX module, the heater is interfaced to an IBM-PC that runs a real time (QNX) operating system. The GPC and identification algorithms (both coded in C) are executed under an in-house developed software called MULTICON. The MULTICON software is tailored to handle any SISO/MIMO control and identification algorithms and also permits configuration of different control strategies. The outlet temperature for this experiment is regulated by manipulating the steam valve position. The stirrer and the baffle in the tank help to maintain a uniform temperature. For this experiment the water level in the tank is maintained constant by a separate pneumatic controller.

The process is suitably excited as a first step in the experiment to identify a model and thereby obtain an estimate of the MPM through signal processing techniques. An open loop process response for square type excitation is shown in Figure 5.31. This information is used to select robust tuning parameters, which are then implemented on the plant. A MATLAB-based toolbox has been developed that utilizes experimental data for robust design of the GPC. The different sets of GPC tuning parameters that were tried on the plant are shown in Table 5.15.

Figure 5.32 shows that for case-1 the stability bound intersects the upperbound and almost intersects the actual uncertainty thus resulting in a marginally stable system as can be seen in Figure 5.33. During the step down in the setpoint, the oscillations in the controller are not observed because the valve reaches its lower saturation limit, thus exhibiting the inherent nonlinearity in the system. The $C_c(q)$ filter in case-2 adds to the robustness in the system by increasing the robustness margin. When NU is set equal to 2, for case-3, it adds to the aggressiveness of the controller and thereby limits the robustness of the controller settings, as shown in the figure. The contribution of λ to the enhancement of the robustness in the system can be seen for case-4, where the controller action is heavily detuned and the robustness margin is comparatively large.

5.16 Estimation of MPM from the Closed-loop data

An important assumption in the robust design of GPC using the SGT criteria is that an ***a priori*** knowledge of the MPM (spectrum) is known. In Chapter 2 it was shown how the signal processing methods could be used to estimate the spectrum of MPM from open-loop process data. The applicability of the robust tuning guidelines for GPC discussed in the earlier sections, therefore remains confined to the: (a) design stage and (b) time invariant plants; when the estimated MPM is based on the open-loop data. In other words, should the plant dynamics change in course of closed-loop control, the existing tuning parameters may no longer guarantee robustness to the modelling errors.

In order to ensure robustness of the controller for changing plant dynamics, it becomes necessary to determine the MPM from closed loop data. The scope of the proposed robust design of GPC therefore gets vastly enhanced when MPM is estimated from the closed loop data, which is the focus of discussion in this section. The derivation of MPM from the closed-loop data are followed by a discussion on the scope and limitations of the proposed method.

The knowledge of MPM can be extracted from the error signal which is designated as $e(t)$ in Figure 5.2. The signal $e(t)$ in Figure 5.2 can be written as:

$$e(t) = R(q)w(t) - S(q)y(t) \tag{5.82}$$

For a noise-free plant, the output $y(t)$ is:

$$y(t) = \frac{e(t)[\hat{G}(q) + \tilde{G}(q)]}{\Delta T(q)} \tag{5.83}$$

Substituting $y(t)$ from equation-(5.83) in equation-(5.82) gives:

$$e(t) = \frac{\Delta T(q)R(q)}{\Delta T(q) + S(q)\underbrace{[\hat{G}(q) + \tilde{G}(q)]}_{G(q)}}w(t) \tag{5.84}$$

By taking $S(q)$ common, the above equation-(5.84) can also be written as:

$$\frac{1}{e(t)} = \frac{S(q)}{\Delta T(q)R(q)w(t)} \left\{ \frac{\Delta T(q) + S(q)\hat{G}(q)}{S(q)} + \frac{S(q)\tilde{G}(q)}{S(q)} \right\} \tag{5.85}$$

Using equation-(5.46), the above relation can be re-expressed as:

$$\begin{aligned} e(t) &= \left(\frac{R(q)\Delta T(q)}{S(q)}\right) \left\{ \frac{M(q)}{1 + M(q)\tilde{G}(q)} \right\} w(t) \\ &= \left(\frac{R(q)\Delta T(q)}{\Delta T(q) + \hat{G}(q)S(q)}\right) \left(\frac{1}{M(q)\tilde{G}(q)}\right) w(t) \end{aligned} \tag{5.86}$$

Further it is easy to show that: $1 - M(q)\hat{G}(q) = \Delta T(q)/(\Delta T(q) + \hat{G}(q)S(q))$, which on substituting in equation-(5.86) gives:

$$e(t) = \underbrace{\frac{1 - M(q)\hat{G}(q)}{1 + M(q)\tilde{G}(q)} R(q)}_{\mathcal{X}(q)} \, w(t) \tag{5.87}$$

In the context of control relevant identification, Rivera *et al.* have derived an expression similar to the above equation-(5.87) in terms of the multiplicative uncertainty $\tilde{G}_m(q)$ and the sensitivity function [31]. The above equation-(5.87) however is in terms of the mixed sensitivity $M(q)$ and $\tilde{G}(q)$ which is used to determine $\tilde{G}(q)$ from the closed loop data (equation-(5.87) is revisited again in Chapter 7, in the context of control relevant identification). Signal processing methods discussed in Chapter 2 can then be applied to equation-(5.87) to obtain a spectral estimate of $\mathcal{X}(q)$ as shown below:

$$\mathcal{X}(\omega) = \frac{\Phi_{ew}(\omega)}{\Phi_{ww}(\omega)} \tag{5.88}$$

From equation-(5.88), it is clear that external excitation (*e.g.* in the set-point $w(t)$) is required in order to obtain a good estimate of $\mathcal{X}(\omega)$. Once $\mathcal{X}(\omega)$ is known then, equation-(5.87) can be used to estimate $\tilde{G}(\omega)$ as:

$$\tilde{G}(\omega) = \frac{[1 - M(\omega)\hat{G}(\omega)]R(\omega) - \mathcal{X}(\omega)}{M(\omega)\mathcal{X}(\omega)} \tag{5.89}$$

If the plant is corrupted by noise: $H(q)\xi(t)$, (which often is the case), then the output $y(t)$ becomes:

$$y(t) = H(q)\xi(t) + \frac{\hat{G}(q) + \tilde{G}(q)}{\Delta T(q)} e(t) \tag{5.90}$$

Substituting $y(t)$ from equation-(5.90) in equation-(5.82) and using: $1 - M(q)\hat{G}(q) = \Delta T(q)/(\Delta T(q) + \hat{G}(q)S(q))$, gives:

$$\frac{1}{e(t)} = \frac{1}{S(q)} \left\{ \frac{1 + M(q)\tilde{G}(q)}{1 - M(q)\hat{G}(q)} \right\} \left\{ \frac{S(q)}{R(q)w(t)} - \frac{1}{H(q)\xi(t)} \right\} \tag{5.91}$$

As in case of equation-(5.87), the above equation-(5.91) can be simplified to:

$$\frac{1}{e(t)} = \frac{1}{\mathcal{X}(q)} \frac{1}{w(t)} - \frac{R(q)}{S(q)H(q)\mathcal{X}(q)} \frac{1}{\xi(t)} \tag{5.92}$$

From equation-(5.92), $1/\mathcal{X}(\omega)$ can be determined as:

$$\frac{\Phi_{e'w'}(\omega)}{\Phi_{w'w'}(\omega)} = \frac{1}{\mathcal{X}(\omega)} - \frac{R(\omega)}{S(\omega)H(\omega)\mathcal{X}(\omega)} \frac{\Phi_{\xi'w'}(\omega)}{\Phi_{w'w'}(\omega)} \tag{5.93}$$

where $e' = 1/e$, $w' = 1/w$ and $\xi' = 1/\xi$. It is common to have $e(t) = 0$ and $w(t) = 0$ under the closed loop operation, hence the use of equation-(5.93) is not recommended as it involves taking inverses of $w(t)$ and $e(t)$. This situation is avoided by rearranging equation-(5.92) to:

$$S(q)H(q)\xi(t)e(t) = S(q)H(q)\mathcal{X}(q)w(t)\xi(t) + R(q)w(t)e(t) \tag{5.94}$$

which leads to:

$$\frac{\Phi_{ew}(\omega)}{\Phi_{ww}(\omega)} = \mathcal{X}(\omega) + \frac{R(q)}{S(q)H(q)} \frac{\Phi_{we}(\omega)\Phi_{w\xi}(\omega)}{\Phi_{\xi\xi}(\omega)\Phi_{ww}(\omega)} \tag{5.95}$$

In presence of noise $v(t) = H(q)\xi(t)$, the use of signal processing methods to estimate $\mathcal{X}(\omega)$ from equation-(5.95) will be biased by:

$$\left\{ \frac{R(q)}{S(q)H(q)} \right\} \frac{\Phi_{we}(\omega)\Phi_{w\xi}(\omega)}{\Phi_{\xi\xi}(\omega)\Phi_{ww}(\omega)}$$

The effect of this bias term is minimal since $w(t)$ is uncorrelated with $\xi(t)$. However for a finite data record (which usually is the case) the effect of $\Phi_{\xi w}(\omega)$ does not vanish completely and from equation-(5.95) it is seen that that if $\Phi_{ww}(\omega)$ rolls off at the higher frequencies, it would add bias in the estimated $\mathcal{X}(\omega)$ at the higher frequencies. The effect of this bias term can be minimized by providing white noise excitation through $w(t)$.

Estimation of Model from the GPC closed loop data

For GPC, the sensitivity function $\mathcal{S}(q)$ is given by:

$$\mathcal{S}(q) = \frac{\Delta T(q)}{\Delta T(q) + S(q)G(q)} \tag{5.96}$$

which can be used to express $e(t)$ as:

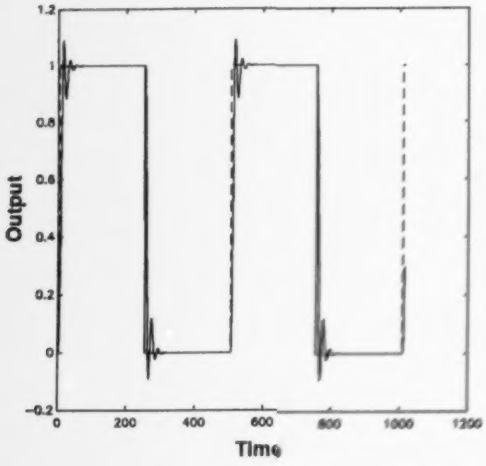


Figure 5.34: Noise-free closed loop output data for step changes in the setpoint.

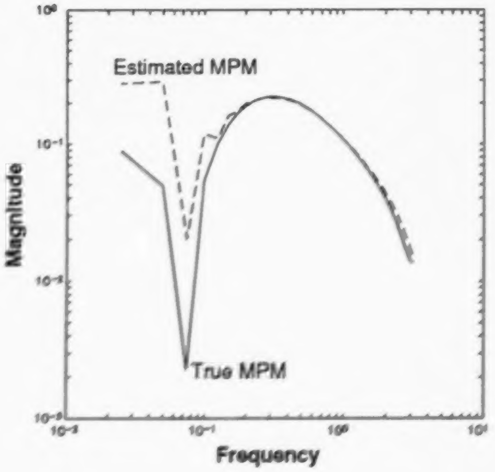


Figure 5.35: Estimated MPM from the noise-free closed loop data.

$$e(t) = \mathcal{S}(q)(R(q)w(t) - S(q)H(q)\xi(t))$$

or

$$e^f(t) = \mathcal{S}(q)w^f(t) - \mathcal{S}(q)H(q)\xi(t) \tag{5.97}$$

where $e^f(t) = e(t)/S(q)$ and $w^f(t) = R(q)w(t)/S(q)$. The method of PEM can be used to estimate $\mathcal{S}(q)$ from equation-(5.97), and $G(q)$ in-turn can then be estimated from $\mathcal{S}(q)$. The error signal $e(t)$ has a poor SNR compared to $y(t)$, therefore $G(q)$ is estimated using:

$$y^f(t) = G(q)w^{ff}(t) + H(q)\xi(t) \tag{5.98}$$

using PEM, where $y(t)/\mathcal{S}(q)$ and $w^{ff}(t) = R(q)w(t)/\Delta T(q)$.

The procedure for the estimation of MPM and model from closed loop data is illustrated by simulation examples in Figures 5.34 to 5.39. The closed loop outputs for GPC shown in Figures 5.34 and 5.36 are obtained for the system given by equations-(5.51) and (5.52). The output in Figure 5.34 corresponds to the noise-free plant and it is obtained by subjecting the plant to step changes in the setpoint. The output shown in Figure 5.36 is obtained by subjecting the plant to white noise excitation through $w(t)$ and here the plant is assumed to be corrupted by white noise signal of SNR$\approx$ 3. In both these cases, 1024 data points are collected, although for the sake of clarity only a portion of the data are shown in Figure 5.36. Equations-(5.88), (5.89) and equation-(5.95) are used to obtain the

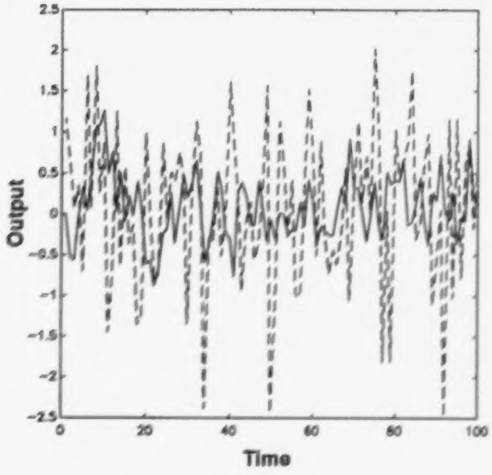


Figure 5.36: A section of closed loop data corrupted with white measurement noise for white noise excitation in the setpoint.

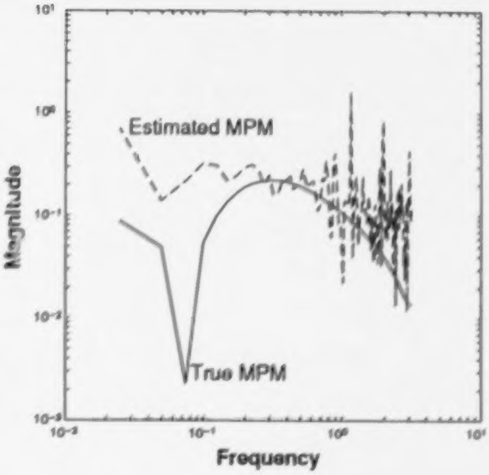


Figure 5.37: Estimated MPM from the noise corrupted closed loop data.

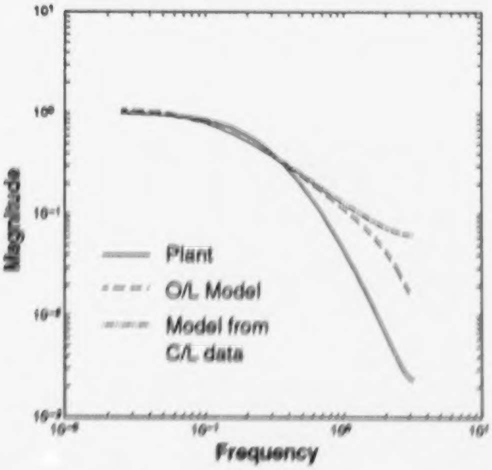


Figure 5.38: Estimated model from the noise-free closed loop data.

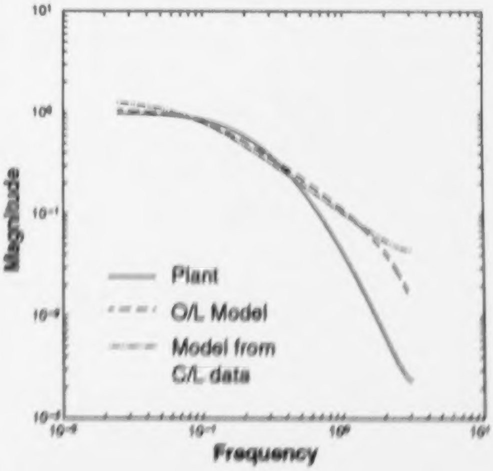


Figure 5.39: Estimated model from the closed loop data corrupted by the measurement noise.

estimated MPM from the closed loop data shown in Figures 5.34 and 5.36. The results of the estimated MPMs are shown in Figures 5.35 (for the noise-free case) and 5.37 (for the noisy case). In both these cases 'Hanning' windows with data overlapping (as discussed in Chapter 2) are used to obtain the MPM spectrums. Square wave excitation through the setpoint results in a good estimation of MPM for the noise-free process, whereas for the noisy process, a white noise excitation in the setpoint is required to obtain an acceptable estimate of MPM.

The closed loop data in Figures 5.34 and 5.36 can also be used to estimate the process model by using equations-(5.96) to (5.98). Figures 5.38 and 5.39 compare the estimated models corresponding to the noise-free and the noisy plants. Overall the estimated model using the closed loop data appear to match well with the model that was estimated using the open loop data. There are however some mismatchs between the open loop and the closed loop based models at the higher frequencies.

Since MPM can be estimated from the closed-loop data, it becomes possible to check the robustness of the existing (GPC) controller on-line. This way any gradual changes in the plant dynamics can be detected and the GPC (or for any LRPC control law where MPM can be estimated) can be appropriately re-tuned on-line based on the SGT criteria.

The main limitation of the proposed method is that sufficiently rich excitation must be provided in the set-point $w(t)$ in order to obtain a good estimate of the MPM or model. It is also assumed that the plant dynamics remain time-invariant during the window of time when closed loop data is collected for identification. Thus the application of this method is quasi-adaptive and it can give reliable estimates only when the changes in the plant dynamics are slower than the time-length of the data window considered. Nevertheless, this method advances the robust design concepts from non-adaptive to quasi-adaptive applications.

5.17 Selection of tuning parameters using an optimization technique

Like most LRPCs, GPC has a large number of tuning parameters. These tuning parameters no doubt add to the flexibility in the design of GPC, but just the sheer number of these tuning parameters can sometimes be bothersome for some users. Ideally one would therefore like to have these tuning parameters selected by using some optimization procedure.

The use of an optimization technique to select all the tuning parameters of GPC appears to be a non-trivial task, because these tuning parameters involve both integer and continuous variables and further these variables are related in a nonlinear way to $|1/M(\omega)|$. For example, equations-(5.41) and (5.42) show that, even for the simple case of a first order model and for $NU = 1$, the controller and model parameters are related in a complex

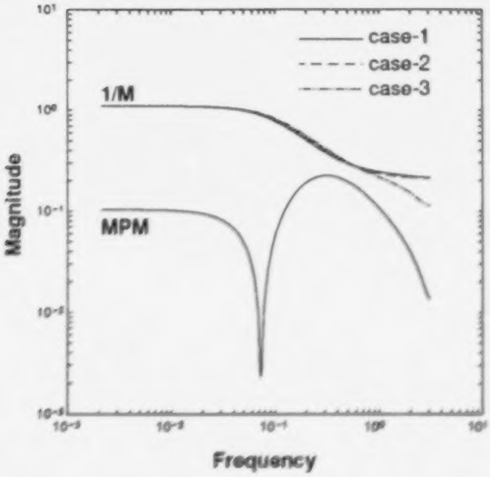


Figure 5.40: Robustness bounds for three optimal sets of tuning parameters for $\epsilon = 0.1$.

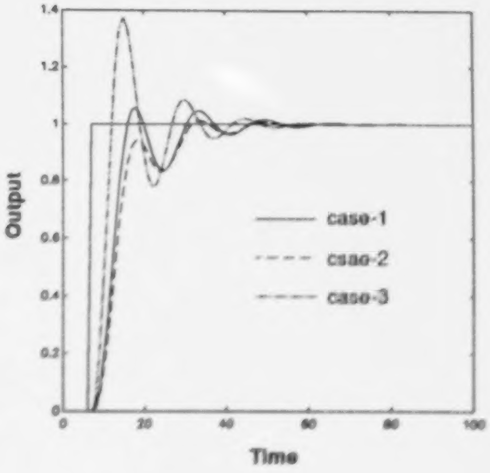


Figure 5.41: Servo responses for three optimal sets of tuning parameters for $\epsilon = 0.1$.

manner.

MINLP[9] can be used to solve optimization[10] problems that involve both integer and continuous variables; but in MINLP the integer variables must be expressed in binary form *e.g.* o or 1. In the context of GPC, the use of binary form of integer tuning parameters to express the controller is beyond the scope of this thesis, hence the focus here is on utilizing only the continuous variables to formulate the optimization problem.

The next issue of importance is the appropriate formulation of the optimization problem in order to obtain tuning (continuous) parameters that give robust stability and performance. The small gain theorem as such is an excellent tool to obtain robust stability, but it cannot give an estimate of the achieved or actual performance. In fact the achieved performance can only be accessed through actual implementation of the controller on the plant. Nevertheless the SGT plots and the corresponding servo responses in Sections 5.7 to 5.12 showed that:

1. As $|1/M(\omega)|$ moves away from the MPM, the controller provides more robustness but performance becomes more detuned and sluggish. Therefore maintaining a large robustness margin is not expected to give a good (achieved) performance.

2. When $|1/M(\omega)|$ is very close to or touches the MPM, then the process response shows

---

[9] MINLP stands for Mixed Integer Non-Linear Programming.

[10] The author would like to thank Dr.R.K.Wood, Department of Chemical Engineering, University of Alberta for discussing this optimization problem.

significant oscillations. Hence close proximity or contact between $|1/M(\omega)|$ and the MPM should be avoided in the interest of stability and performance.

3. The performance is usually acceptable when $|1/M(\omega)|$ is neither too far nor too close to the MPM. It is difficult to quantify how far or how close $|1/M(\omega)|$ should be from the MPM.

The above observations roughly indicate that good performance with a guaranteed robustness can be expected by *minimizing* the area between $|1/M(\omega)|$ and MPM (*i.e.* the robustness margin), subject to the SGT constraints and by maintaining a user choosen minimum robustness margin. Thus the optimization problem (to select the tuning parameters) can be stated as:

$$
\begin{aligned}
& \vartheta = \mathrm{Min} \textstyle\sum_{k=1}^{n_w} \Delta\omega_k(|\frac{1}{M(\omega_k)}| - |\tilde{G}(\omega_k)|) \\
\text{subject to:} & \\
& \tilde{G}(\omega_k) + \epsilon < |\tfrac{1}{M(\omega_k)}| \text{ for } k = 1, \ldots, n_w \\
& \underline{\lambda} \leq \lambda \leq \overline{\lambda} \\
& 0 \leq c \leq 1 \\
& 1 \leq p \leq \overline{p} \\
& 0 \leq \gamma_\infty \leq 1
\end{aligned}
\tag{5.99}
$$

where in the above equation-(5.99), n_w is the number of frequency points, ϵ is the user specified minimum robustness margin at any frequency, $\{\underline{\lambda}, \overline{\lambda}\}$ are the lower and upper bounds on the controller weighting λ, c is a parameter in the filter $C_c(q) = 1 - cq^{-1}$ (only the first order filter is considered) and $\overline{p}$ is the upper bound for p in $P(q) = p - (p-1)q^{-1}$. The parameter ϵ in equation-(5.99) serves to indirectly specify the performance criterian.

The use of this optimization procedure (equation-(5.99)) to obtain the GPC parameters is illustrated by simulation examples for the system given by equations-(5.51) and (5.52). Table 5.17 gives a layout of different case studies and their results are shown in Table 5.17. Sequential quadratic programming (SQP), under MATLAB's Optimization Tool Box was used to obtain these tuning parameters. The robustness bounds and performance corresponding to the optimal tuning parameters (in Table 5.17) are illustrated in Figures 5.40 to 5.43; and they indicate the following:

- The λ values generally do not get selected, which (based on earlier analysis) is a desirable outcome. For the case of $NU = 2$, the λ values selected are marginally small and they can be neglected. The λ values do not get selected, probably because it causes a 'minimum' in $|1/M(\omega)|$ (which implies more robustness margin at the higher frequencies).

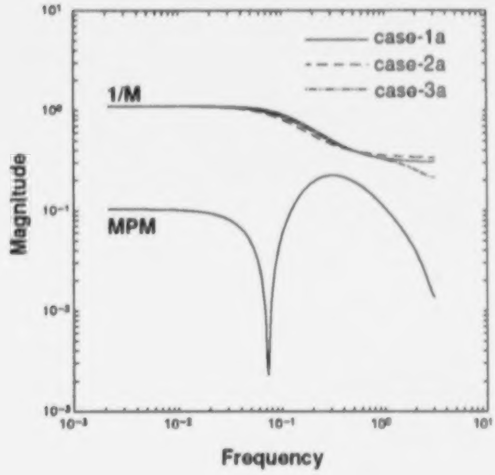


Figure 5.42: Robustness bounds for three optimal sets of tuning parameters for $\epsilon = 0.2$.

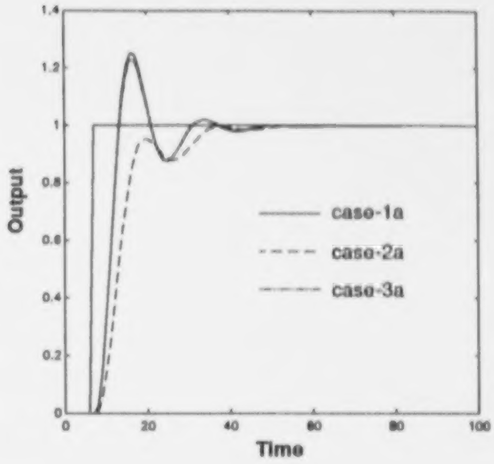


Figure 5.43: Servo responses for three optimal sets of tuning parameters for $\epsilon = 0.2$.

- The optimization program puts significant emphasis on $C_c(q)$ and $P(q)$ to adjust the robustness margins. Expectedly for smaller values of ϵ, $C_c(q)$ selected is small, but it is difficult to explain the selection criteria for $P(q)$. For $NU = 2$, $C_c(q)$ selected is large in order to compensate for the smaller robustness margin due to the use of $NU = 2$.
- Interestingly, γ_∞ is not selected for cases-1 and 1a (*i.e.* $N_2 = 5$), whereas it gets selected for cases-2 and 2a (where $N_2 = 10$).
- The robustness margins for cases-1 and 2 and for cases-1a and 2a are very close and they look alike, but their corresponding servo responses are significantly different. Thus it is difficult to get an estimate of the achieved performance from the SGT based robustness bound plots.
- The proposed optimization method gives optimal tuning parameters such that they are robust to the modelling errors and satisfy a certain minimum specified robustness margin.

5.18 Conclusions

The main contributions of this chapter are as follows:

- A systematic and complete controller synthesis technique has been provided that combines signal processing techniques for the estimation of MPM (which is presented in Chapter 2), predictive controller tuning and stability analysis via the small gain theorem.

- Using the small gain theorem it is shown that the stability of GPC cannot be guaranteed if the steady state gain of the uncertainty exceeds the model gain. At steady state, the small gain condition will never be violated if the steady state gain of the model exceeds half the steady state gain of the process.
- The robustness bound of GPC (*i.e.* $|1/M(\omega)|$) is influenced by the model at lower frequencies. At higher frequencies the robustness bound is influenced more by the controller term $|\Delta T(\omega)/S(\omega)|$, provided that the model rolls off these higher frequencies.
- Explicit expressions for computing the GPC Diophantine coefficients, GPC controller law and linear GPC polynomials are derived for a process represented by a first order model with a delay. The use of these GPC expressions can hasten the computational speed of GPC in the context of adaptive or non-adaptive control.
- The influence of GPC tuning parameters such as N_2, $C_c(q)$, $P(q)$, λ and γ_∞ on the stability and robustness has been examined. It has been shown through simulations that, with nominal increases in the values of all of these parameters, the robustness bound $|1/M(\omega)|$ increases, thus giving robustness to the system. The fact that NU makes the controller more active has also been verified through simulation results analysed in the frequency domain.
- It is shown analytically that the increase in the values of the tuning paramaters such as N_2, λ and $C_c(q^{-1}) = 1 - cq^{-1}$, causes an increase in the magnitude spectrum of the inverse of the GPC controller transfer function (*i.e.* $|\Delta T(\omega)/S(\omega)|$) when applied to a first order model of the process.
- It is analytically proved that with the increase in the values of the tuning parameters such as N_2, λ and $C_c(q^{-1}) = 1 - cq^{-1}$ there is also an increase in the robustness margin. The change in the magnitude spectrum of the robustness margin is more pronounced in the higher frequency range than in the lower frequency range.
- The robust stability results have been evaluated experimentally on a pilot-scale heater system. The experimental results show that this method can be implemented on an open-loop stable process.
- A method is presented to estimate MPM from the closed loop plant data, therefore the robustness of GPC can be examined quasi-adaptively.
- An optimization method is used to automatically select some of the tuning parameters of GPC, such that robust performance is guaranteed.

Table 5.1: Different cases of GPC parameters for disturbance rejection.

Case	N_1	N_2	NU	λ	$P(q)$	$C_c(q)$
1	1	5	1	0	1	$1-0.8q^{-1}$
2	1	5	1	1	1	$1-0.8q^{-1}$
3	1	15	1	0	1	$1-0.8q^{-1}$
4	1	5	1	0	1	$[1-0.8q^{-1}]^2$
5	1	5	1	0	1	$1-0.8q^{-1}$

Table 5.2: Different cases of GPC parameters for the experimental set-up.

Case	N_1	N_2	NU	λ	$P(q)$	$C_c(q)$
1	1	5	1	0	1	1
2	1	5	1	0	1	$1-0.8q^{-1}$
3	1	5	2	0	1	$1-0.8q^{-1}$
4	1	5	1	0.5	1	$1-0.8q^{-1}$

Table 5.3: Layout of different cases studies to determine optimal tuning parameters.

$\epsilon = 0.1$				$\epsilon = 0.2$			
	case-1	case-2	case-3		case-1a	case-2a	case-3a
N_2	5	10	10	N_2	5	10	10
NU	1	1	2	NU	1	1	2

Table 5.4: Small gain theorem based optimal tuning parameters of GPC.

	λ	$C_c(q)$	$P(q)$	γ_∞	ϑ	Iterations
case-1	0	$1-0.6629q^{-1}$	$4.7857-3.7857q^{-1}$	0	0.6281	61
case-1a	0	$1-0.9342q^{-1}$	$1.0040-0.0040q^{-1}$	0	0.9234	46
case-2	0	$1-0.6030q^{-1}$	$5.0000-4.0000q^{-1}$	1	0.6118	91
case-2a	0	$1-0.7427q^{-1}$	$5.0000-4.0000q^{-1}$	1	0.9736	76
case-3	0.0001	$1-0.9006q^{-1}$	$2.4428-1.4428q^{-1}$	0.0067	0.4841	46
case-3a	0.0096	$1-0.9117q^{-1}$	$3.4603-2.4603q^{-1}$	0.0007	0.7910	157

Bibliography

[1] G. Zames. "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems. Part-I: Conditions Using Concepts of Loop Gain, Conicity and Positivity". *IEEE Trans. on AC*, 11(2):228–238, 1966.

[2] C.A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties.* Academic Press, New York, USA, 1975.

[3] R.L. Kosut. "Adaptive uncertainty modelling: on-line robust control design". In *American Control Conference*, pages 245–250, Minneapolis, USA, 1987.

[4] B. Wahlberg and L. Ljung. "Hard frequency-domain model error bounds from least-squares like identification techniques". *IEEE Trans. Automatic Control*, 37:900–912, 1992.

[5] D.J. Cloud and B. Kouvaritakis. "Statistical bounds on multivariable frequency response: an extension of the generalized Nyquist criterion". *IEE proceed. pt-D*, 133(3):97–110, 1986.

[6] R.C. Younce and C.E. Rohrs. "Identification with non-parametric uncertainty ". In *Proceedings of IEEE Conf. on Decision and Control*, pages 3154–3161, Honolulu, USA., 1990.

[7] E. Goberdhansingh, L. Wang, and W.R. Cluett. "Robust Control system design using direct frequency response ". In *Proceedings of American Control Conf.*, pages 3026–3030, Chicago, USA., 1992.

[8] R.L. Kosut, M.K. Lau, and S.P. Boyd. "Set-membership identification of systems with parametric and nonparametric uncertainty". *IEEE Trans. in Automatic Control*, 37:929–941, 1992.

[9] G.C. Goodwin, M. Gevers, and B. Ninness. "Quantifying the error in estimated transfer functions with application to model order selection". *IEEF Trans. in Automatic Control*, 37:913–928, 1992.

[10] G.M. Jenkins and D.G. Watts. *Spectral Analysis and its Application.* Holden-Day, San Franscisco, USA, 1968.

[11] A.V. Oppenheim, A.S. Willsky, and I.T. Young. *Signals and Systems.* Prentice Hall of India Pvt. Ltd, New Delhi, India, 1987.

[12] Kay. S.M. and S.L. Marple. "Spectrum Analysis - A Modern Perspective". *Proceedings of the IEEE*, 69(11):1380 – 1418, 1981.

[13] C.R. Cutler and R.B. Hawkins. "Application of large predictive controller to a hydrocracker second stage reactor". In *Proceedings of American Control Conf.*, pages 284–291, Atlanta, USA., 1988.

[14] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. "Generalized Predictive Control-Part I and II". *Automatica*, 23:137–160, 1987.

[15] D.W. Clarke. "Application of generalized predictive control to industrial processes". *IEEE Control Syst. Mag.*, 8:49–55, 1988.

[16] C.E. Garcia, D.M. Prett, and M. Morari. "Model Predictive Control: Theory and practice - a survey ". *Automatica*, 25:335–348, 1989.

[17] R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control - The Thinking Man's GPC.* Prentice Hall International Series., 1990.

[18] B.D. Robinson and D.W. Clarke. "Robustness effects of a pre filter in GPC". *IEE Proceedings-D*, 138:2 – 8, 1991.

[19] D.E. Seborg, T.F. Edgar, and D.A. Mellichamp. *Process Dynamics and Control.* John Wiley & Sons, USA, 1989.

[20] D.W. Clarke and C. Mohtadi. "Properties of generalized predictive control". *Automatica*, 25:859–875, 1989.

[21] C.H. Mohtadi. *Studies in Advanced Self-tuning Algorithm.* Ph.D. Thesis, University of Oxford, Oxford, U.K., 1986.

[22] K. Kwok. *Long-Range Adaptive Predictive Control: Extensions and Applications.* Ph.D. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1992.

[23] K.Y. Kwok and S.L. Shah. "Long-Range Predictive Control with a Terminal Matching Condition". *Chem. Eng. Sci.*, 49(9):1287 – 1300, 1994.

[24] M. Saudagar. *Unified Predictive Control.* PhD. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1995.

[25] A.R. McIntosh. *Performance and Tuning of Adaptive Generalized Predictive Control.* MSc. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1988.

[26] A.R. McIntosh, S.L. Shah, and D.G. Fisher. "Analysis and Tuning of Adaptive GPC". *Can. J. of Chem. Engg.*, 69:97–110, 1991.

[27] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. *Feedback Control Theory.* Macmillan Publishing Co., New York, USA, 1992.

[28] B. Kouvaritakis and I. Postlethwaite. "Principal gains and phases: insensitive robustness measures for assessing the closed-loop property". *IEE Proc., Pt-D*, 129:233–241, 1982.

[29] C. Mohtadi. *On the role of pre-filtering in parameter estimation and control.* Springer-Verlag, Berlin, Germany, 1988. In S.L.Shah and G.Dumont (Eds), Adaptive Control Strategies for Industrial Use.

[30] L. Ljung. *System Identification-Theory For The User.* Prentice Hall, New Jersey, USA, 1987.

[31] D.E. Rivera, J.F. Pollard, and C.E. Garcia. "Control-Relevant Prefiltering: A Systematic Design Approach and Case Study". *IEEE Trans. on AC*, 37(7):964 – 974, 1992.

Chapter 6

Robustness of Markov-Laguerre model based Predictive Controller

A SISO linear model predictive controller with steady state weighting is formulated using a Markov-Laguerre model. A structured noise model is combined with the Markov-Laguerre model to achieve faster disturbance rejection. The small gain theorem is used to analyze robustness of the Markov-Laguerre based controller in the presence of unmodelled dynamics. It is shown that robustness properties of the Markov-Laguerre based controller and GPC are similar.

6.1 Introduction

The concept of robust design of GPC discussed in chapter 5 is extended in this chapter to design a model predictive controller (MPC) based on the Markov-Laguerre model.

In Chapters 3 and 4 it was shown that orthonormal models such as Laguerre models can result a fairly accurate descriptions of the plant dynamics even in presence of significant measurement noise. Furthermore, delays, inverse response or other unusual high frequency dynamics which are common in chemical processes can be effectively captured using Markov-Laguerre or Markov-Kautz models as described earlier in Chapter 4. Although this chapter is concerned only with the robust design of Markov-Laguerre model based MPC, it is straight forward to extend these methods for the design of predictive controller based on the Markov-Kautz model (for processes with complex poles).

Furthermore, the Markov-Laguerre model is unstructured like the FIR/step-response models. Therefore the robust design procedure discussed in this chapter can easily be applied to other MPCs such as DMC and IDCOM that are based on FIR/step-response models. The Markov-Laguerre based MPC has several advantages over other well established controllers such as DMC/IDCOM. Notable among them is that the Markov-Laguerre model requires a smaller number of parameters compared to the large number of coefficients required by the FIR/step-response models to emulate the process behavior; and yet the Markov-Laguerre model retains the same structure as the FIR/step-response models. Many commercial controllers such as IDCOM/DMC still prefer to use FIR/step response models because these models are well suited for capturing unusual high frequency dynamics and are easily interpretted by process engineers. The Markov-Laguerre model is effective even in such cases because unusual high frequency process behavior can easily be captured using the Markov (or FIR) parameters whereas the lower frequency dynamics can be effectively modelled using the Laguerre component. Since the Markov-Laguerre model requires a smaller number of coefficients, its usage can lead to savings in computer resouces. This saving in computer resources can be even more significant for large scale integrated control systems involving large dimensional MIMO systems.

The use of a Laguerre model to design unconstrained MIMO predictive controller was first proposed by Zervos and Dumont [1]. With respect to the Laguerre model based unconstrained SISO MPC, Dumont and co-workers showed that: (a) the design of such controllers is always stable for a unity controller horizon and (b) under some parametric mismatch conditions, the robustness of such controllers can be established [2]. Dumont and co-workers also reported several successful industrial applications of such Laguerre based (adaptive) MPC [2, 3, 4, 5].

Subsequently Finn *et al.* [6] proposed the use of Markov-Laguerre models to design a constrained MIMO predictive controller whose formulation is similar to QDMC[1] [6]. For Markov-Laguerre based MPC, Finn *et al.* have: (a) indicated that the choice of tuning parameters is important in order to ensure robustness to modelling errors; and (b) showed from the passivity theory that input constraints do not destabilize the loop [6].

Some of the work carried out by Dumont and co-workers and Finn *et al.* in formulating orthonormal model based MPC is extended in this chapter. Section 6.2 discusses the conversion of Markov-Laguerre model to state-space form, which is subsequently used to design the MPC. The state-space model used by Dumont and co-workers is represented in terms of the parameters of the Laguerre model in the continuous domain, whereas the state-space model used in this chapter is in terms of the parameters of the Markov-Laguerre model that is identified in the discrete domain. Finn *et al.* have also used the state-space equivalent of discrete Markov-Laguerre model, but they did not elaborate on their state-space model in the way presented in this chapter. This section also provides an explicit expression for an ARX model that can be obtained from the Markov-Laguerre model.

Section 6.3 describes the use of Markov-Laguerre based state-space model to design a MPC. The approach followed here to design the MPC is similar to the Laguerre based MPC that was formulated by Dumont *et al.* [3]. The concept of steady-state weighting as proposed by Kwok and Shah for GPC [7] is also used here to formulate Markov-Laguerre based MPC.

Section 6.4 presents the formulation of constrained Markov-Laguerre based MPC which is very much similar to what has been proposed by Finn *et al.* [6]. But this section also provides explicit expressions for the constraints that are of practical interest.

---

[1] QDMC stands for Quadratic Dynamic Matrix Control.

Dumont and co-workers did not consider any noise model for their Lagurre based MPC. However Finn *et al.* considered a noise model of the type $1/\Delta^n$ (their analysis was however based on $n = 1$) to design Markov-Laguerre based MPC [6]. This idea of noise model is extended in Section 6.5 to design Markov-Laguerre MPC that is based on $C(q)/D(q)\Delta$ type noise model. Thus a combination of structured noise model and unstructured Markov-Laguerre based model is combined to obtain better process regulation.

Section 6.6 discusses the design of a Markov-Laguerre based MPC that is robust to the unstructured or non-parametric modelling errors. The of different tuning parameters on the robustness of the Markov-Laguerre based MPC are examined through simulation case studies in Section 6.7. A comparative study between GPC and Markov-Laguerre based MPC with and without the use of noise model is also discussed in a subsection under the Section 6.7. The discussions presented in the preceeding sections are followed by concluding remarks in Section 6.8.

6.2 Markov-Laguerre based state-space model

The discrete Markov-Laguerre model as introduced in Chapters 3 and 4 is expressed as:

$$\hat{G}(z) = \sum_{k=1}^{d} h_k z^{-k} + \sum_{k=1}^{N} g_k L_k(z) z^{-d} \tag{6.1}$$

to represent the dynamics of a discrete overdamped stable linear plant $G(z)$ by its estimated model $\hat{G}(z)$; where

$$L_k(z) = \frac{\sqrt{1-a^2}}{z-a}\left(\frac{1-az}{z-a}\right)^{k-1} \tag{6.2}$$

is the Laguerre basis function for a specified real pole a. In equation-(6.1), N is the order of the Laguerre model, d is the delay or length of the inverse response, $h_k, (k = 1, 2, \ldots, d)$ are the Markov coefficients and $g_k, (k = 1, 2, \ldots, N)$ are the Laguerre coefficients. The model output corresponding to equation-(6.1) is given by:

$$\hat{y}(t) = \sum_{k=1}^{d} h_k u(t-k) + \sum_{k=1}^{N} g_k l_k(t-d) \tag{6.3}$$

where the Laguerre basis signal is: $l_k(t-d) = L_k(q)u(t-d)$.

With the **bold** faced upper and lower case characters denoting matrices and vectors respectively, the discrete states-space model:

$$\mathbf{x}(t+1) \;=\; \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t)$$

$$\hat{y}(t) \;=\; \mathbf{c}^T\mathbf{x}(t) \tag{6.4}$$

can be used to represent the model given by equation-(6.3) where

$$\mathbf{x}(t) = [u(t-1), u(t-2), \ldots, u(t-d), l_1(t-d), l_2(t-d), \ldots, l_N(t-d)]^T \in \Re^{n\times 1}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \in \Re^{n\times n}$$

$$\mathbf{b} = [1, \mathbf{0}_{1\times(N+d-1)}]^T \in \Re^{n\times 1}$$

$$\mathbf{c} = [h_1, h_2, \ldots, h_d, g_1, g_2, \ldots, g_N]^T \in \Re^{n\times 1} \tag{6.5}$$

such that $n = N + d$, $\mathbf{A}_{11} \in \Re^{d\times d}$, $\mathbf{A}_{12} \in \Re^{d\times N}$, $\mathbf{A}_{21} \in \Re^{N\times d}$, $\mathbf{A}_{22} \in \Re^{N\times N}$ and where:

$$\begin{aligned}
\mathbf{A}_{11} &= \begin{bmatrix} \mathbf{0}_{1\times d} & \\ \mathbf{I}_{d-1} & \mathbf{0}_{(d-1)\times 1} \end{bmatrix} \\
\mathbf{A}_{12} &= \mathbf{0}_{d\times N} \\
\mathbf{A}_{21} &= \begin{bmatrix} \mathbf{0}_{1\times(d-1)} & \alpha \\ \mathbf{0}_{1\times(d-1)} & -a\alpha \\ \vdots & \vdots \\ \mathbf{0}_{1\times(d-1)} & -a^{N-1}\alpha \end{bmatrix} \\
\mathbf{A}_{22} &= \left[\begin{array}{cccccc} a & & & & & \\ \alpha^2 & a & & & & \\ -a\alpha^2 & \alpha^2 & a & & & \\ a^2\alpha^2 & -a\alpha^2 & \alpha^2 & a & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ (-a)^{N-2}\alpha^2 & (-a)^{N-3}\alpha^2 & \ddots & \ddots & \ddots & a \end{array}\right]
\end{aligned} \tag{6.6}$$

In the above expression (6.6) $\alpha = \sqrt{1-a^2}$, $\mathbf{I}_p$ is identity matrix of size p and $\mathbf{0}_{p\times q}$ is a matrix of zeros of size $p \times q$. For the special case of a Laguerre model (*i.e.* when $d = 0$), the state matrix reduces to $\mathbf{A} = \mathbf{A}_{22}$, $\mathbf{b}$ becomes the last column of $\mathbf{A}_{21}$ and $\mathbf{c} = [g_1, g_2, \ldots, g_N]^T$. The Markov-Laguerre model given by equation-(6.1) can also be expressed in the ARX form as:

$$\begin{aligned}
(1-az^{-1})^N y(t) &= \{(1-az^{-1})^N(h_1 + h_2z^{-1} + \ldots + h_dz^{-d+1}) \\
&+ \alpha z^{-d}[g_1(1-az^{-1})^{N-1} + g_2(1-az^{-1})^{N-2}(z^{-1}-a) \\
&+ g_3(1-az^{-1})^{N-3}(z^{-1}-a)^2 + \ldots + g_N(z^{-1}-a)^{N-1}]\}u(t-1)
\end{aligned} \tag{6.7}$$

6.3 Design of unconstrained MPC - without noise model

As discussed in Chapter 5, MPCs are designed to mimimize the sum of square of the error between the projected set-point and the predicted output over a prediction horizon from N_1 to N_2 with a penalty λ on the controller movement Δu over a control horizon NU. The objective function to be minimized is:

$$\underset{\min\ \Delta\mathbf{u}}{J} = [\mathbf{w} - \mathbf{y}^P]^T \Gamma_y [\mathbf{w} - \mathbf{y}^P] + \Delta\mathbf{u}^T \mathbf{\Lambda} \Delta\mathbf{u} \tag{6.8}$$

where $\mathbf{w} \in \Re^{(N_2 - N_1 + 1) \times 1}$ is the projected setpoint vector, $\Delta\mathbf{u} \in \Re^{NU \times 1}$ is the projected incremental input, $\Lambda = \lambda \mathbf{I}_{NU}$ and $\mathbf{y}^P \in \Re^{(N_2 - N_1 + 1) \times 1}$ is the predicted output which is given by:

$$\mathbf{y}^P = [\hat{y}(t + N_1|t), \hat{y}(t + N_1 + 1|t), \ldots, \hat{y}(t + N_2|t)]^T \tag{6.9}$$

Successive substitution of equation-(6.4) gives the following i^{th} step predicted output as [2]:

$$\hat{y}(t + i|t) = \mathbf{c}^T \mathbf{A}^i \mathbf{x}(t) + \nu_{i-1} u(t) + \nu_{i-2} u(t + 1) + \cdots + \nu_0 u(t + i - 1) \tag{6.10}$$

where $\nu_i = \mathbf{c}^T \mathbf{A}^i \mathbf{b}$. The set of equations-(6.10) for $i = 1$ to N_2 can then be written as:

$$\begin{bmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \vdots \\ \hat{y}(t+NU|t) \\ \hat{y}(t+NU+1|t) \\ \vdots \\ \hat{y}(t+N_2|t) \end{bmatrix}_{N_2 \times N_2} = \begin{bmatrix} \mathbf{c}^T\mathbf{A} \\ \mathbf{c}^T\mathbf{A}^2 \\ \vdots \\ \mathbf{c}^T\mathbf{A}^{NU} \\ \mathbf{c}^T\mathbf{A}^{NU+1} \\ \vdots \\ \mathbf{c}^T\mathbf{A}^{N_2} \end{bmatrix}_{N_2 \times n} \mathbf{x}(t)$$

$$+ \begin{bmatrix} \nu_0 & & & & & & \\ \nu_1 & \nu_0 & & & & & \\ \vdots & & \ddots & & & & \\ \nu_{NU-1} & \cdots & \cdots & \nu_0 & & & \\ \nu_{NU} & \cdots & \cdots & \cdots & \nu_0 & & \\ \vdots & & & & & \ddots & \\ \nu_{N_2-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \nu_0 \end{bmatrix}_{N_2 \times N_2} \begin{bmatrix} u(t) \\ u(t+2) \\ \vdots \\ u(t+NU-1) \\ u(t+NU) \\ \vdots \\ u(t+NU) \end{bmatrix}_{N_2 \times 1} \tag{6.11}$$

Using the relation: $s_i = \sum_{k=0}^{i} \nu_k$, the above equation-(6.11) can be rearranged as:

$$\mathbf{y}^P = \underbrace{\mathbf{L}'\mathbf{x}(t) + \mathbf{s}_1 u(t-1)}_{\mathbf{f}'} + \mathbf{S}\Delta\mathbf{u} \tag{6.12}$$

for the prediction horizon N_1 to N_2 where:

$$\begin{aligned}
\mathbf{L}' &= \left[\mathbf{c}^T\mathbf{A}^{N_1}, \mathbf{c}^T\mathbf{A}^{N_1+1}, \ldots, \mathbf{c}^T\mathbf{A}^{N_2}\right]^T \\
\Delta\mathbf{u} &= [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+NU)]^T \\
\mathbf{S} &= \begin{bmatrix} s_{N_1} & & \\ s_{N_1+1} & s_{N_1} & \\ \vdots & \vdots & \ddots \\ s_{N_2-N_1+1} & \cdots & \cdots \end{bmatrix}_{(N_2-N_1+1)\times NU}
\end{aligned} \tag{6.13}$$

In equation (6.12), $\mathbf{f}' \in \Re^{(N_2-N_1+1)\times 1}$ corresponds to the free-response, $\mathbf{s}_1 \in \Re^{(N_2-N_1+1)\times 1}$ is the first column of $\mathbf{S} \in \Re^{(N_2-N_1+1)\times NU}$, s_i in $\mathbf{S}$ is the i^{th} step response coefficient and $\mathbf{L}' \in \Re^{(N_2-N_1+1)\times(N+d)}$ is the observability matrix. $\mathbf{S}\Delta u$ corresponds to the forced response. Substituting equation-(6.13) in (6.8) results in the following control law:

$$\Delta\mathbf{u} = (\mathbf{S}^T\mathbf{\Gamma}_y\mathbf{S} + \mathbf{\Lambda})^{-1}\mathbf{S}^T\mathbf{\Gamma}_y\mathbf{e}' \tag{6.14}$$

where

$$\mathbf{\Gamma}_y = \begin{bmatrix} \gamma_{y1} & & & \\ & \gamma_{y2} & & \\ & & \ddots & \\ & & & \gamma_{yN_2} \end{bmatrix} \tag{6.15}$$

and where the projected error is $\mathbf{e}' = \mathbf{w} - \mathbf{f}' \in \Re^{(N_2-N_1+1)\times 1}$. The term $\mathbf{e}'$ is however devoid of the feedback from output $y(t)$, therefore as noted by Elshafei *et al.* [2], $\mathbf{e}'$ is replaced by $\mathbf{e} = \mathbf{w} - \mathbf{f}$ where the free-response $\mathbf{f}$ is:

$$\mathbf{f} = \mathbf{y} + \mathbf{L}\mathbf{x}(t) + \mathbf{s}_1 u(t-1) \tag{6.16}$$

where

$$\begin{aligned}
\mathbf{y} &= [y(t), y(t), \ldots, y(t)]^T \in \Re^{(N_2+N_1-1)\times 1} \\
\mathbf{L} &= [\mathbf{c}^T(\mathbf{A}^{N_1} - \mathbf{I}), \mathbf{c}^T(\mathbf{A}^{N_1+1} - \mathbf{I}), \mathbf{c}^T(\mathbf{A}^{N_2-N_1+1} - \mathbf{I})]^T \in \Re^{(N_2-N_1+1)\times n}
\end{aligned} \tag{6.17}$$

6.3.1 Steady State Weighting - γ_∞

As mentioned in Chapter 5, Kwok and Shah modified the LRPC objective by augmenting the GPC control law with a terminal matching condition [7]. The modified LRPC objective with a steady state weighting γ_∞ gives the effect of a large prediction horizon without actually using a large value for N_2. For the case of GPC it has been shown that the steady state weighting γ_∞ endows better stability properties than the move supression factor λ [8]. Interestingly, this result appears to be somewhat similar (though not quite the same) to the guaranteed stability properties of infinite horizon receding controller [9, 10].

Since γ_∞ improves the stability of GPC, it becomes meaningful to explore the effect of γ_∞ on the stability of Markov-Laguerre model based MPC as well. The key step in including the steady state weighting is in the formulation of predictor at the infinite horizon (*i.e.* at steady state) which is given by the following two *lemmas*:

Lemma 6.1 *The predicted output for the Markov-Laguerre model at steady state is given by:*

$$y^P(t+\infty|t) = y(t) - \mathbf{c}^T\mathbf{I}\mathbf{x}(t) + s_\infty u(t-1) + \mathbf{s}_\infty \Delta\mathbf{u} \tag{6.18}$$

where $s_\infty = \sum_{k=0}^{\infty} \nu_k$ *and* $\mathbf{s}_\infty = [s_\infty, \cdots, s_\infty]_{1\times NU}$

Proof: This *lemma* is proved by evaluating the predictor $\mathbf{y}^P$ given by a combination of equations-(6.10), (6.12), (6.16) and (6.17) at $i = \infty$ and noting that $\mathbf{A}^\infty = 0$. □

Lemma 6.2 s_∞ *in equation-(6.18) is given by:*

$$s_\infty = \mathbf{c}^T\mathbf{A}_\infty\mathbf{b} \tag{6.19}$$

where for the Markov-Laguerre model, $\mathbf{A}_\infty$ *is*

$$\mathbf{A}_\infty = \begin{bmatrix} \mathbf{A}_{\infty 11} & \mathbf{0}_{d\times(N-d)} \\ \mathbf{A}_{\infty 21} & \mathbf{A}_{\infty 22} \end{bmatrix} \tag{6.20}$$

(for the Laguerre model, $\mathbf{A}_\infty = \mathbf{A}_{\infty 22}$*),* $\mathbf{A}_{\infty 11} \in \Re^{d\times d}$ *is a lower triangular matrix of ones (including the diagonal) i.e.:*

$$\mathbf{A}_{\infty 11} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & \cdots & \cdots & 1 \end{bmatrix} \tag{6.21}$$

all elements of $\mathbf{A}\infty_{21} \in \Re^{(N-d)\times d}$ *in equation-(6.20) are* $\alpha/(1-a)$ *(a and* α *are defined in (6.2 and 6.6)) and* $\mathbf{A}\infty_{22} \in \Re^{(N-d)\times(N-d)}$ *is given by:*

$$\mathbf{A}_{\infty 22} = \begin{bmatrix} \frac{1}{1-a} & & & & \\ \frac{1+a}{1-a} & \frac{1}{1-a} & & & \\ \frac{1+a}{1-a} & \frac{1+a}{1-a} & \frac{1}{1-a} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \frac{1+a}{1-a} & \cdots & \cdots & \frac{1+a}{1-a} & \frac{1}{1-a} \end{bmatrix} \tag{6.22}$$

Proof. Using the relations $\nu_i = \mathbf{c}^T\mathbf{A}^i\mathbf{b}$ and $s_i = \sum_{k=0}^{i}\nu_k$, $\mathbf{A}_\infty$ in equation-(6.20) can be obtained as:

$$\mathbf{A}_\infty = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^\infty \tag{6.23}$$

Substituting equation-(6.6) in equation-(6.23) proves the *lemma*. This *lemma* is also proved by using the following relation:

$$\mathbf{A}_\infty = [\mathbf{I} - \mathbf{A}]^{-1} \tag{6.24}$$

because at steady state, $z = 1$ in the transfer function: $\mathbf{c}^T[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}$. □

The objective function with a steady state weighting γ_∞ is given by [7]:

$$J = [\mathbf{w} - \mathbf{y}^P]^T\mathbf{\Gamma}_y[\mathbf{w} - \mathbf{y}^P] + \Delta\mathbf{u}^T\mathbf{\Lambda}\Delta\mathbf{u} + [w_\infty - {y_\infty}^P]^T\gamma_\infty[w_\infty - {y_\infty}^P] \tag{6.25}$$

where w_∞ is the setpoint at the steady state which is assumed to be same as the elements in $\mathbf{w}$. Substituting equation-(6.18) in (6.25) and minimizing it with respect to $\Delta\mathbf{u}$ gives the following control law with a steady state weighting γ_∞ term:

$$\begin{aligned} \Delta\mathbf{u} &= \mathbf{H}[\mathbf{\Gamma}_y\mathbf{S}^T(\mathbf{w} - \mathbf{f}) + \gamma_\infty\mathbf{S}_\infty^T(w_\infty - f_\infty)] \\ \text{where} \qquad \mathbf{H} &= [\mathbf{S}^T\mathbf{\Gamma}_\mathbf{y}\mathbf{S} + \mathbf{\Lambda} + \mathbf{S}_\infty^T\gamma_\infty\mathbf{S}_\infty]^{-1} \\ \mathbf{S}_\infty &= \begin{bmatrix} s_\infty & & \\ \vdots & \ddots & \\ s_\infty & \cdots & s_\infty \end{bmatrix} \end{aligned} \tag{6.26}$$

Saudagar [11] has shown that the steady state s_∞ is not affected by NU, hence it does not make sense to use the $\mathbf{S}_\infty$ matrix as shown in equation-(6.26). A simplified control law with steady state weighting can therefore be obtained by forming an augmented predicted output as:

$$\mathbf{y}_\mathbf{a}^P = [[\mathbf{y}^P]^T, y^P(t + \infty|t)]^T \tag{6.27}$$

and then minimizing the objective function with respect to $\Delta\mathbf{u}$:

$$J = \text{Min} \quad [\mathbf{w}_a - \mathbf{y}_a^P]^T \boldsymbol{\Gamma} [\mathbf{w}_a - \mathbf{y}_a^P] + \Delta\mathbf{u}^T \boldsymbol{\Lambda} \Delta\mathbf{u} \tag{6.28}$$

where $\mathbf{w}_a = [\mathbf{w}^T, w_\infty]^T$ and

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_y & \\ & \gamma_\infty \end{bmatrix} \tag{6.29}$$

to give the following simplified controller law:

$$\Delta\mathbf{u} = \mathbf{H}_a[\boldsymbol{\Gamma}\mathbf{S}_a^T(\mathbf{w}_a - \mathbf{f}_a)] \tag{6.30}$$

where

$$\begin{aligned}
\mathbf{H}_a &= [\mathbf{S}^T\boldsymbol{\Gamma}\mathbf{S} + \boldsymbol{\Lambda}]^{-1} \\
\mathbf{S}_a &= \begin{bmatrix} \mathbf{S} & \\ s_\infty & \mathbf{0}_{1\times NU-1} \end{bmatrix} \\
\mathbf{f}_a &= \mathbf{y} + \underbrace{\begin{bmatrix} \mathbf{L} \\ -\mathbf{I} \end{bmatrix}}_{\mathbf{L}_a} \mathbf{x}(t) + \underbrace{\begin{bmatrix} \mathbf{s}_1 \\ s_\infty \end{bmatrix}}_{\mathbf{s}_a} u(t-1)
\end{aligned} \tag{6.31}$$

6.4 Design of constrained MPC

In a plant the final control elements are always bounded by their saturation limits and their incremental movements are often constrained due to physical reasons. Such constriants are accomadated in the MPC by formulating the following problem:

$$\begin{array}{lcc}
J^+ & \overset{\min}{=\Delta u} \ \tfrac{1}{2}\Delta\mathbf{u}^T\mathbf{H}_a\Delta\mathbf{u} - \mathbf{g}_a^T\Delta\mathbf{u} & \\
\min\Delta\mathbf{u} & & \\
\text{s.t.} & & \mathbf{C}\Delta\mathbf{u} \leq \mathbf{p}
\end{array} \tag{6.32}$$

where

$$\begin{aligned}
\mathbf{H}_a &= [\mathbf{S}_a^T\boldsymbol{\Gamma}\mathbf{S}_a + \boldsymbol{\Lambda}]^{-1} \\
\mathbf{g}_a &= \boldsymbol{\Gamma}\mathbf{S}_a^T(\mathbf{w}_a - \mathbf{f}_a)
\end{aligned} \tag{6.33}$$

Quadratic programming (QP) can be used to solve this problem, where the resulting J^+ is a suboptimal solution of J (*i.e.* $J^+ > J$) given by equation-(6.30). In equation-(6.32), the output constraints are handled by expressing $\mathbf{y}^P$ in terms of $\Delta\mathbf{u}$. The limitations on the control inputs represent the 'hard' input constraints which are never violated. On the other hand the output constraints are 'soft' which are often violated due to the presence of noise, disturbances and unmodelled dynamics.

Let $\{\overline{u}, \underline{u}\}$, $\{\overline{\Delta u}, \underline{\Delta u}\}$ and $\{\overline{\Delta \mathbf{y}}, \underline{\Delta \mathbf{y}}\}$ denote the maximum and minimum saturation values of the input, incremental input and projected incremental output around the setpoint $\mathbf{w}$ repectively. In view of these constraints, the operating regime becomes:

$$\begin{aligned} \underline{u} \leq & \quad u(t+i) & \leq \overline{u} & \quad \text{for } i = 0, \ldots NU-1 \\ \underline{\Delta u} \leq & \quad \Delta \mathbf{u} & \leq \overline{\Delta u} & \\ \underbrace{\mathbf{w} + \underline{\Delta \mathbf{y}}}_{\underline{\mathbf{y}}} \leq & \quad \mathbf{y}^P & \leq \underbrace{\mathbf{w} + \overline{\Delta \mathbf{y}}}_{\overline{\mathbf{y}}} & \end{aligned} \tag{6.34}$$

where each element of $\underline{\mathbf{y}}$ is $\underline{y}$ and each element of $\overline{\mathbf{y}}$ is $\overline{y}$. It is mathematically possible to have different bounding limits on $u(t+i)$ and $\Delta \mathbf{u}$ over the controller horizon NU. However it is practical to assume constant bounding limits on $\{u(t+i), \Delta \mathbf{u}\}$ over the controller horizon because the hard input constraints usually remain constant over time and even if they change, they are expected to change gradually and it is difficult to predict these changes in the future. With these assumptions, the constraint matrices $\mathbf{C}$ and $\mathbf{p}$ in equation-(6.32) becomes:

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} \mathbf{I}_{NU} & \mathbf{C}_1 & -\mathbf{I}_{NU} & -\mathbf{C}_1 & \mathbf{S} & -\mathbf{S} \end{bmatrix}^T \\ \mathbf{p} &= \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 & \mathbf{p}_5 & \mathbf{p}_6 \end{bmatrix}^T \end{aligned} \tag{6.35}$$

where $\mathbf{C}_1 \in \Re^{(NU-1)\times NU}$ has the following structure for $NU > 2$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & \cdots & 1 & 0 \\ 1 & \cdots & \cdots & 1 \\ \vdots & & & \vdots \\ 1 & \cdots & \cdots & 1 \end{bmatrix} \tag{6.36}$$

and for $NU = 2$, $\mathbf{C}_1 = [1, 1]$. In equation-(6.35)

$$\begin{aligned} \mathbf{p}_1 &= [\min(\overline{\Delta u}, \overline{u} - u(t-1)), \overline{\Delta u}, \ldots, \overline{\Delta u}]^T \in \Re^{NU\times 1} \\ \mathbf{p}_2 &= [\overline{u} - u(t-1), \ldots, \overline{u} - u(t-1)]^T \in \Re^{(NU-1)\times 1} \\ \mathbf{p}_3 &= [\max(-\underline{\Delta u}, -(\underline{u} - u(t-1))), -\underline{\Delta u}, \ldots, -\underline{\Delta u}]^T \in \Re^{NU\times 1} \\ \mathbf{p}_4 &= [-(\underline{u} - u(t-1)), \ldots, -(\underline{u} - u(t-1))]^T \in \Re^{(NU-1)\times 1} \\ \mathbf{p}_5 &= [\underbrace{w(N_1|t) + \overline{\Delta y}}_{\overline{y}} - f(N_1|t), \ldots, w(N_2|t) + \overline{\Delta y} - f(N_2|t)]^T \in \Re^{(N_2-N_1+1)\times 1} \end{aligned}$$

$$\mathbf{p}_6 = [-(\underbrace{w(N_1|t) - \underline{\Delta y}}_{\underline{y}} - f(N_1|t)), \ldots, -(w(N_2|t) - \underline{\Delta y} - f(N_2|t))]^T \in \Re^{(N_2-N_1+1)\times 1} \tag{6.37}$$

As an illustration, for $NU = 2$ we have (neglecting the effect of γ_∞):

$$\begin{array}{rcl} \left.\begin{array}{r} \underline{u} \leq u(t) \leq \overline{u} \\ \underline{u} \leq u(t+1) \leq \overline{u} \end{array}\right\} & \Rightarrow & \begin{array}{c} \underline{u} - u(t-1) \leq \Delta u(t) \leq \overline{u} - u(t-1) \\ \underline{u} - \Delta u(t) \leq \Delta u(t+1) + u(t-1) \leq \overline{u} - \Delta u(t) \end{array} \\ \underline{\Delta u} \leq & \Delta u(t) & \leq \overline{\Delta u} \\ \underline{\Delta u} \leq & \Delta u(t+1) & \leq \overline{\Delta u} \\ \underline{\mathbf{y}} - \mathbf{f} \leq & \mathbf{S}\Delta\mathbf{u} & \leq \overline{\mathbf{y}} - \mathbf{f} \end{array} \tag{6.38}$$

the above set of equations-(6.38) will then yield:

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ -1 & -1 \\ s_1 & 0 \\ s_2 & s_1 \\ \vdots & \vdots \\ s_{N_2} & s_{N_2-1} \\ -s_1 & 0 \\ -s_2 & -s_1 \\ \vdots & \vdots \\ -s_{N_2} & -s_{N_2-1} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \end{bmatrix}}_{\Delta\mathbf{u}} = \underbrace{\begin{bmatrix} \min(\overline{\Delta u}, \overline{u} - u(t-1)) \\ \overline{\Delta u} \\ \overline{u} - u(t-1) \\ \max(\underline{\Delta u}, \underline{u} - u(t-1)) \\ -\underline{\Delta u} \\ -(\underline{u} - u(t-1)) \\ \overline{y} - f_{N_1} \\ \overline{y} - f_{N_1+1} \\ \vdots \\ \underline{y} - f_{N_2} \\ -(\underline{y} - f_{N_1}) \\ -(\underline{y} - f_{N_1+1}) \\ \vdots \\ -(\underline{y} - f_{N_2}) \end{bmatrix}}_{\mathbf{P}} \tag{6.39}$$

6.5 Design of unconstrained MPC - with a noise model

The modelling of noise plays an important role in the formulation of some MPCs such as GPC or UPC [12, 11]. The noise model influences both robustness and performance of the controller; for example, in Chapter 5 it was shown how the use of $C_c(q)$ filter improves the robustness of GPC.

Noise models are an integral part of structured models such as ARIMAX or Box-Jenkins (BJ) models. Consequently, the noise model naturally gets considered in the derivation of MPCs that are based on such structured models. The noise model acts as an observer for the anticipated disturbances, thus it enables the MPC to take suitable corrective actions in advance to obtain better regulation. The inclusion of noise model also helps to account for the modelling errors as exhibited by GPC or UPC [12, 11].

Traditionally the unstructured models such as the step-response or orthonormal models are represented without any consideration for the noise model. Therefore noise models usually get excluded in the formulation of MPCs that are based on unstructured models. Consequently the disturbance rejection properties of unstructured model based MPCs such as classical DMC are poor. Finn *et al.* combined a noise model of type $1/\Delta^n$ (the analysis was however based on $n = 1$) with a Markov-Laguerre model to formulate the MPC [6]. Finn *et al.* used the noise model from the identification perspective and they did not elaborate on the consequence of including the noise model in the controller performance. This section presents a slightly different formulation for the Markov-Laguerre based MPC by considering the following noise model:

$$v(t) = \frac{C(q)}{D(q)\Delta}\xi(t) = \frac{1 + cq^{-1}}{\Delta(1 + dq^{-1})}\xi(t) \tag{6.40}$$

The type of noise model shown by equation-(6.40) has also been used by Saudagar [11] to obtain a faster disturbance rejection for the DMC; where the DMC in that case was based on the transfer function model. However here the structured model has been combined with the state-space form to obtain a controller design. The noise model in this case is restricted to a first order, because subsequently it will become apparent that the derivation of the Markov-Laguerre based MPC becomes quite involved even for this case of first order noise model. The noise model considered here is to be viewed from both the identification and control perspectives. Both these perspectives are important because the knowledge of the noise model helps to identify a better quality model and it can lead to better regulatory response. The process model under consideration thus has the following form:

$$y(t) = \frac{B(q)}{A(q)}u(t-1) + \frac{C(q)}{D(q)\Delta}\xi(t) \tag{6.41}$$

Rearranging the above equation gives:

$$y^f(t) = \frac{B(q)}{A(q)}u^f(t-1) + \xi(t) \tag{6.42}$$

where $y^f(t) = D(q)\Delta y(t)/C(q)$ and $u^f(t-1) = D(q)\Delta u(t-1)/C(q)$. The transfer function term $B(q)/A(q)$ in equation-(6.42) can also be identified as Markov-Laguerre model, in such a case the process model becomes:

$$y^f(t) = \{\sum_{i=1}^{d} h_i q^{-1} + \sum_{i=1}^{N} g_i q^{-d} L_i(q)\}u^f(t) + \xi(t) \tag{6.43}$$

The model shown by equation-(6.43) is a result of combination of the unstructured Markov-Laguerre model and a structured noise model. The Markov-Laguerre model shown by equation-(6.43) can be estimated from the process data $\{y(t), u(t)\}$ using a batch least-squares (LS) technique. If the noise model in equation-(6.40) is unavailable, then it can be approximated from the residual data $r(t) = y(t) - \hat{y}(t)$, where $\hat{y}(t)$ is the model output corresponding to the identified Markov-Laguerre model. One way to identify the noise model from $r(t)$ is to apply an AR model[2] of arbitrarily high order to $r(t)$ to estimate the innovations $\hat{\xi}(t)$ as [13]:

$$\hat{\xi}(t) = (1 - \Upsilon(q))r(t) \tag{6.44}$$

where $\Upsilon(q)$ represents the identified estimated high order AR parameters. Subsequently $D(q)$ and $C(q)$ can be estimated by forming the following ARX model [13]:

$$\Delta D(q)r(t) = [C(q) - 1]\hat{\xi}(t) + \xi(t) \tag{6.45}$$

The estimated noise model using this two step approach is influenced by the modelling errors as the residual $r(t)$ also contains information on the MPM. The model described by equation-(6.43) can be expressed in the state-space form as:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u^f(t) \\ \mathbf{y}^f(t) &= \mathbf{c}^T\mathbf{x}(t) \end{aligned} \tag{6.46}$$

where $\mathbf{x}(t) = [u^f(t-1), \ldots, u^f(t-d), l_1^f(t-d), \cdots, l_N^f(t-d)]$, $l_k^f(t-d) = L_k(q)u^f(t-d)$ and

$$\begin{aligned} y^f(t) &= \Delta y(t) + d\Delta y(t-1) - cy^f(t-1) \\ u^f(t) &= \Delta u(t) + d\Delta u(t-1) - cu^f(t-1) \end{aligned} \tag{6.47}$$

Note 6.1 *The difference between c in $C(q)$ in equation-(6.40) and $\mathbf{c}$ in equation-(6.46)) should be noted.*

Using equation-(6.47), the i^{th} step output prediction is given by:

$$\Delta y(t+i|t) = y^f(t+i) + cy^f(t+i-1) - d\Delta y(t+i-1) \tag{6.48}$$

For 1 to N_2 step ahead predictions, equation-(6.48) together with equation-(6.46) can be expressed as:

---

[2] The command 'AR(r,N)' can be used to obtain the AR parameters using MATLAB for the residual vector r and AR-order N.

$$
\begin{aligned}
\underbrace{\begin{bmatrix} y(t+1|t)\\ y(t+2|t)\\ \vdots\\ y(t+N_2|t)\end{bmatrix}}_{\mathbf{Y}^P} &= \underbrace{(1-d)\begin{bmatrix} y(t|t)\\ y(t+1|t)\\ \vdots\\ y(t+N_2-1|t)\end{bmatrix} + d\begin{bmatrix} y(t-1)\\ y(t|t)\\ \vdots\\ y(t+N_2-2|t)\end{bmatrix}}_{\mathbf{Y}'}\\
&+ \underbrace{\mathbf{c}^T(\mathbf{A}+c\mathbf{I})}_{\mathbf{r}}\begin{bmatrix} \mathbf{x}(t)\\ \mathbf{x}(t+1)\\ \vdots\\ \mathbf{x}(t+N_2-1)\end{bmatrix} + \mathbf{c}^T\mathbf{b}\begin{bmatrix} u^f(t)\\ u^f(t+1)\\ \vdots\\ u^f(t+N_2-1)\end{bmatrix}
\end{aligned}
\tag{6.49}
$$

Successive substitution of $\mathbf{x}(t)$ from equation-(6.46) in equation-(6.49) results in

$$
\begin{aligned}
\mathbf{Y}^P &= \mathbf{Y}' + \mathbf{r}\left\{\underbrace{\begin{bmatrix} \mathbf{I}\\ \mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{N_2-1}\end{bmatrix}}_{\tilde{\mathbf{L}}}\mathbf{x}(t) + \begin{bmatrix} \mathbf{0}_{n\times 1} & & & \\ \mathbf{b} & & & \\ \mathbf{A}\mathbf{b} & \mathbf{b} & & \\ \vdots & \ddots & \ddots & \\ \mathbf{A}^{N_2-2}\mathbf{b} & \mathbf{A}^{N_2-3}\mathbf{b} & \cdots & \mathbf{b}\end{bmatrix}\times\right.\\
&\left.\begin{bmatrix} u^f(t)\\ u^f(t+1)\\ u^f(t+2)\\ \vdots\\ u^f(t+N_2-1)\end{bmatrix}\right\} + \mathbf{c}^T\mathbf{b}\begin{bmatrix} u^f(t)\\ u^f(t+1)\\ u^f(t+2)\\ \vdots\\ u^f(t+N_2-1)\end{bmatrix}\\
&= \mathbf{Y}' + \mathbf{r}\tilde{\mathbf{L}}\mathbf{x}(t) + \underbrace{\begin{bmatrix} \mathbf{c}^T\mathbf{b} & & & & \\ \mathbf{r}\mathbf{b} & \mathbf{c}^T\mathbf{b} & & & \\ \mathbf{r}\mathbf{A}\mathbf{b} & \mathbf{r}\mathbf{b} & \mathbf{c}^T\mathbf{b} & & \\ \vdots & \ddots & \ddots & \ddots & \\ \mathbf{r}\mathbf{A}^{N_2-2}\mathbf{b} & \mathbf{r}\mathbf{A}^{N_2-3}\mathbf{b} & \ddots & \ddots & \mathbf{c}^T\mathbf{b}\end{bmatrix}}_{\mathbf{V}}\times
\end{aligned}
$$

$$\underbrace{\begin{bmatrix} u^f(t) \\ u^f(t+1) \\ u^f(t+2) \\ \vdots \\ u^f(t+N_2-1) \end{bmatrix}}_{\mathbf{u}^f} \tag{6.50}$$

where the elements of $\mathbf{V}$ are:

$$\mathbf{V} = \begin{bmatrix} v_1 & & & \\ v_2 & v_1 & & \\ \vdots & \ddots & \ddots & \\ v_{N_2} & \ddots & \ddots & v_1 \end{bmatrix}_{N_2 \times N_2} \tag{6.51}$$

Recursive substitutions of the predictions $y^P(t+i|t)$ in equation-(6.50) leads to:[3]

$$\mathbf{Y}^P = \boldsymbol{\Xi} y(t) + \boldsymbol{\Psi} y(t-1) + \boldsymbol{\Pi}\mathbf{x}(t) + \boldsymbol{\Theta}\mathbf{u}^f \tag{6.52}$$

where

$$\begin{aligned} \boldsymbol{\Xi} &= \begin{bmatrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \\ \vdots \\ \Xi_{N_2} \end{bmatrix}_{N_2 \times 1} \quad \text{where} \quad \left\{ \begin{array}{l} \Xi_1 = 1 - d \\ \Xi_2 = (1-d)\Xi_1 + d \\ \Xi_3 = (1-d)\Xi_2 + d\Xi_2 \\ \vdots \\ \Xi_{N_2} = (1-d)\Xi_{N_2-1} + d\Xi_{N_2-2} \end{array} \right. \\ \boldsymbol{\Psi} &= \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \vdots \\ \Psi_{N_2} \end{bmatrix}_{N_2 \times 1} \quad \text{where} \quad \left\{ \begin{array}{l} \Psi_1 = d \\ \Psi_2 = (1-d)\Psi_1 \\ \Psi_3 = (1-d)\Psi_2 + d\Psi_1 \\ \vdots \\ \Psi_{N_2} = (1-d)\Psi_{N_2-1} + d\Psi_{N_2-2} \end{array} \right. \end{aligned}$$

---

[3] Length and complexity of equations for the output prediction increases with increase in the prediction horizon N_2. For higher order noise models the expressions for the predictions are much more complicated and they are difficult/tedious to generalize. Hence formulation of Markov-Laguerre based MPC is restricted to a first order noise model.

$$\mathbf{\Pi} = \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \\ \vdots \\ \Pi_{N_2} \end{bmatrix}_{N_2 \times 1} \quad \text{where} \quad \begin{cases} \Pi_1 = \mathbf{rI} \\ \Pi_2 = (1-d)\Pi_1 + \mathbf{rA} \\ \Pi_3 = (1-d)\Pi_2 + d\Pi_1 + \mathbf{rA}^2 \\ \vdots \\ \Pi_{N_2} = (1-d)\Pi_{N_2-1} + d\Pi_{N_2-2} + \mathbf{rA}^{N_2-1} \end{cases}$$

$$\mathbf{\Theta} = \begin{bmatrix} \Theta_{11} & & & & \\ \Theta_{21} & \Theta_{22} & & & \\ \Theta_{31} & \Theta_{32} & \Theta_{33} & & \\ \vdots & \ddots & \ddots & \ddots & \\ \Theta_{N_2,1} & \Theta_{N_2,2} & \ddots & \ddots & \Theta_{N_2 N_2} \end{bmatrix}_{N_2 \times N_2}$$

where

$$\begin{cases} \Theta_{11} = v_1 & & & \\ \Theta_{21} = (1-d)\Theta_{11} + v_2 & \Theta_{22} = v_1 & & \\ \Theta_{31} = (1-d)\Theta_{21} + d\Theta_{11} + v_3 & \Theta_{32} = \Theta_{21} & \Theta_{33} = v_1 & \\ \vdots & \ddots & \ddots & \\ \Theta_{N_2,1} = (1-d)\Theta_{N_2-1,1} + d\Theta_{N_2-2,1} + v_{N_2} & \Theta_{N_2,2} = \Theta_{N_2-1,1} & \ddots & \ddots \end{cases} \tag{6.53}$$

Also by successive substitution of $\mathbf{u}^f$ from equation-(6.46) in conjunction with equation-(6.47) results in:

$$\underbrace{\begin{bmatrix} u^f(t) \\ u^f(t+1) \\ u^f(t+2) \\ \vdots \\ u^f(t+N_2-1) \end{bmatrix}}_{\mathbf{u}^f} = \underbrace{\begin{bmatrix} 1 & & & & \\ d-c & 1 & & & \\ -c(d-c) & (d-c) & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ c^{N_2-2}(d-c) & c^{N_2-3}(d-c) & \cdots & (d-c) & 1 \end{bmatrix}}_{\mathbf{M}_0} \underbrace{\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \\ \vdots \\ \Delta u(t+N_2-1) \end{bmatrix}}_{\Delta\mathbf{u}}$$

$$+ \underbrace{\begin{bmatrix} d \\ -cd \\ c^2 d \\ \vdots \\ -c^{N_2-1} d \end{bmatrix}}_{\mathbf{M}_1} \Delta u(t-1) + \underbrace{\begin{bmatrix} -c \\ c^2 \\ -c^3 \\ \vdots \\ -c^{N_2} \end{bmatrix}}_{\mathbf{M}_2} u^f(t-1) \tag{6.54}$$

Substituting $\mathbf{u}^f$ from the above equation-(6.54) in equation-(6.50) gives:

$$\mathbf{Y}^P = \Xi y(t) + \Psi y(t-1) + \Pi \mathbf{x}(t) + \underbrace{\Theta \mathbf{M}_0}_{\mathbf{S}} \Delta \mathbf{u} + \underbrace{\Theta \mathbf{M}_1}_{\mathbf{\Omega}} \Delta u(t-1) + \underbrace{\Theta \mathbf{M}_2}_{\mathbf{\Phi}} u^f(t-1) \tag{6.55}$$

The control law is obtained by substituting $\mathbf{Y}^P$ from equation-(6.55) in equation-(6.8) and then minimizing the resultant expression with respect to $\Delta \mathbf{u}$ to yield:

$$\Delta \mathbf{u} = \mathbf{H}[\mathbf{S}^T \mathbf{\Gamma}_y \{\mathbf{w} - \Xi y(t) - \Psi y(t-1) - \Pi \mathbf{x}(t) - \mathbf{\Omega} \Delta u(t-1) - \Theta u^f(t-1)\}] \tag{6.56}$$

where $\mathbf{H} = [\mathbf{S}^T \mathbf{\Gamma}_y \mathbf{S} + \mathbf{\Lambda}]^{-1}$.

6.6 Robustness Analysis

In Chapter 5 it was shown how the ***small gain theorem*** [14] can be used to examine the robustness of GPC in the presence of unmodelled dynamics. The robust design method involves estimating the MPM and then designing a controller that is robust to these modelling errors [15]. This method for the robust design of GPC is applied to the Markov-Laguerre based MPC in this section. Infact this method for ascertaining the robustness can be applied to any unstructured model based MPC such as DMC or IDCOM. The present work parallels the work done by Qi [16] in the context of robust design of model predictive controller where the model has a dual form (*i.e.* a combination of FIR and ARX models) and where this model is converted into a state-space form for the controller design. However, in the present case the SGT criterion has been used as a robustness measure whereas the design procedure by Qi is based on the matrix perturbation theory.

Before proceeding to the robustness analysis it is important to check if the designed closed loop system is stable or not. For simplicity, first the MPC which is designed without using any noise model is considered. Using equations-(6.30) and (6.31), the controller (for $\mathbf{w}_a = 0$) can be expressed as:

$$u(t) = -v_1 y(t) - \mathbf{v}_2 \mathbf{x}(t) + v_3 u(t-1)$$

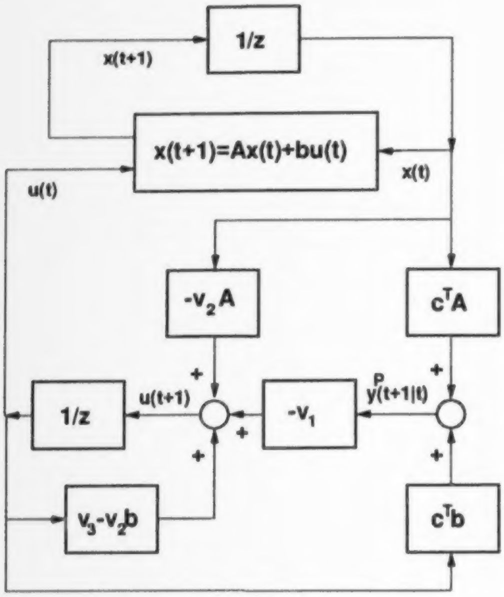


Figure 6.1: Closed loop structure of Markov-Laguerre Predictive Controller

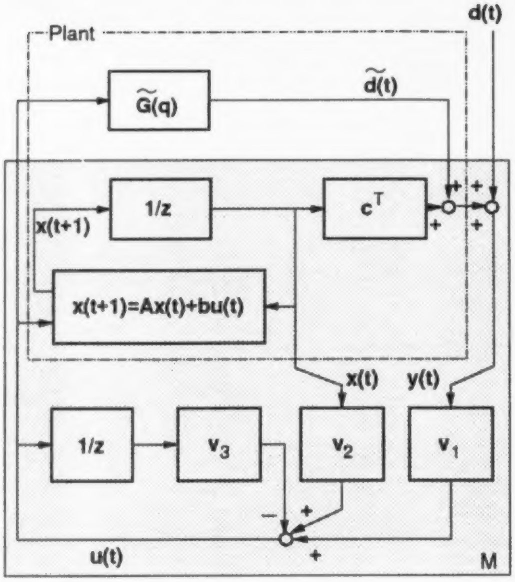


Figure 6.2: Small Gain Structure for the Markov-Laguerre Predictive Controller

$$\begin{aligned} \text{where} \qquad v_1 &= \mathbf{h}_1\mathbf{S}_a^T\boldsymbol{\Gamma}\mathbf{1} \\ \mathbf{v}_2 &= \mathbf{h}_1\mathbf{S}_a^T\boldsymbol{\Gamma}\mathbf{L}_a \\ v_3 &= 1-\mathbf{h}_1\mathbf{S}_a^T\boldsymbol{\Gamma}\mathbf{s}_a \end{aligned} \tag{6.57}$$

where $\mathbf{h}_1$ is the first row of $\mathbf{H}_a$ in equation (6.30) and $\mathbf{1}$ is a vector of ones of size $(N_2+1)\times 1$. The first row of $\mathbf{H}_a$ is only considered because of the receding horizon policy for the MPC. The expression for the designed closed loop is obtained by incrementing $u(t)$ in equation-(6.57) by one-step and substituting $y(t+1|t) = \mathbf{c}^T(\mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t))$ from equation-(6.4) in the resulting incremental equation to give:

$$\begin{bmatrix} \mathbf{x}(t+1) \\ u(t+1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ -(v_1\mathbf{c}^T\mathbf{A}+\mathbf{v}_2\mathbf{A}) & v_3-\mathbf{v}_2\mathbf{b}-v_1\mathbf{c}^T\mathbf{b} \end{bmatrix}}_{\mathbf{A}_c} \begin{bmatrix} \mathbf{x}(t) \\ u(t) \end{bmatrix} \tag{6.58}$$

The designed controller when implemented on the plant will be stable only if $\mathbf{A}_c$ is stable, therefore before proceeding for the robustness analysis it is imperative to check if all the eigenvalues of $\mathbf{A}_c$ are inside the unit circle. Elshafei *et al.* [2] showed that $\mathbf{A}_c$ is stable for Laguerre based MPC when $NU = 1$, but this result does not apply to Markov-Laguerre based MPC as illustrated by examples in later sections.

In Chapter 5, the robustness of GPC was analysed by separating the unmodelled dynamics from the designed closed-loop system. A simple way of representing the MPM is the ***additive purturbation*** (*i.e.* $\tilde{G}(z) = G(z) - \hat{G}(z)$) [17]. Let the corresponding designed closed-loop for $\tilde{G}(\omega)$ be denoted by $M(\omega)$ as shown in Figure 6.2. Such a partitioned system is guaranteed to be stable if $|\tilde{G}(\omega)M(\omega)| < 1$ or $|\tilde{G}(\omega)| < |1/M(\omega)|$ for a SISO system. As in the case of GPC in Chapter 5, here also $|1/M(\omega)|$ is designated as the robustness bound corresponding to $\tilde{G}(\omega)$. Such a method gives a conservative design, nevertheless it results in a controller design which guarantees robustness to unstructured modelling errors.

In Figure 6.2, $\tilde{d}(t)$ represents signal corresponding to the modelling error and the noise-free plant output is $y(t) = \hat{y}(t) + \tilde{d}(t)$, where from equation-(6.4), $\hat{y}(t) = \mathbf{c}^T[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}u(t)$. Combining the expressions for $\hat{y}(t)$ and $y(t)$ with equation-(6.57) gives the following expression for $M(z)$:

$$M(z) = \frac{u(z)}{\tilde{d}(z)} = \frac{-v_1 z}{z\{1 + (\mathbf{v}_2 + v_1\mathbf{c}^T)(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\} - v_3} \tag{6.59}$$

or the robustness bound is given by

$$\begin{aligned}
\left|\frac{1}{M(z)}\right| &= \left|\frac{z\{1 + (\mathbf{v}_2 + v_1\mathbf{c}^T)(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\} - v_3}{-v_1 z}\right| \\
&= \left|\frac{1}{\mathbf{h}_1\mathbf{S}_a^T\mathbf{\Gamma}\mathbf{1}}\left\{\left(\frac{1}{z} - 1\right) - \mathbf{h}_1\mathbf{S}_a^T\mathbf{\Gamma}\left(\frac{\mathbf{s}_a}{z} + \mathbf{L}_a[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}\right)\right\} - \underbrace{\mathbf{c}^T[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}}_{\hat{G}(z)}\right| \\
&= \left|-\left\{\hat{G}(z) + \frac{1}{z}\frac{\mathbf{h}_1\mathbf{S}_a^T\mathbf{\Gamma}}{\mathbf{h}_1\mathbf{S}_a^T\mathbf{\Gamma}\mathbf{1}}\left(\mathbf{s}_a + z\mathbf{L}_a[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b} + \Delta\right)\right\}\right|
\end{aligned} \tag{6.60}$$

The effect of some of the key tuning parameters of the Markov-Laguerre based predictive controller are discussed and compared with the ARIMAX based GPC in the next section. As in the case of GPC, the robustness bound given by equation-(6.60) has the following property:

Lemma 6.3 *At $\omega = 0$ the robustness bound for the Markov-Laguerre based MPC is the steady state gain of the model which is given by:*

$$\left|\frac{1}{M(1)}\right| = \left|\hat{G}(1)\right| = |\mathbf{c}^T[\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}| \tag{6.61}$$

Proof. At steady state $\omega = 0 \Rightarrow z = 1$ and $\Delta = 1 - z^{-1} = 0$ and substituting these in equation-(6.60) gives:

$$\left|\frac{1}{M(1)}\right| = \left| - \left\{ \hat{G}(1) + \frac{\mathbf{h}_1\mathbf{S}_a^T\mathbf{\Gamma}}{\mathbf{h}_1\mathbf{S}_a^T\mathbf{\Gamma}\mathbf{1}} \underbrace{[\mathbf{s}_a + \mathbf{L}_a[\mathbf{I}-\mathbf{A}]^{-1}\mathbf{b}]}_{\mathbf{X}} \right\} \right| \tag{6.62}$$

Representing $\mathbf{s}_a$ in terms of $s_i = \sum_{k=0}^{i} \mathbf{c}^T\mathbf{A}^k\mathbf{b}$ and using the expression for $\mathbf{L}_a$ from equations-(6.17) and (6.31) in the above equation-(6.62) leads to the following form for $\mathbf{X}$:

$$\begin{aligned} \mathbf{X} &= \mathbf{c}^T \begin{bmatrix} \mathbf{I} \\ \mathbf{I}+\mathbf{A} \\ \vdots \\ \mathbf{A}_\infty \end{bmatrix} \mathbf{b} + \mathbf{c}^T \begin{bmatrix} [\mathbf{A}-\mathbf{I}][\mathbf{I}-\mathbf{A}]^{-1} \\ [\mathbf{A}^2-\mathbf{I}][\mathbf{I}-\mathbf{A}]^{-1} \\ \vdots \\ [\mathbf{A}^\infty-\mathbf{I}][\mathbf{I}-\mathbf{A}]^{-1} \end{bmatrix} \mathbf{b} \\ &= \mathbf{c}^T \begin{bmatrix} [\mathbf{A}-\mathbf{I}][\mathbf{I}-\mathbf{A}]^{-1}+\mathbf{I} \\ [\mathbf{A}^2-\mathbf{I}][\mathbf{I}-\mathbf{A}]^{-1}+[\mathbf{I}+\mathbf{A}] \\ \vdots \\ -\mathbf{I}[\mathbf{I}-\mathbf{A}]^{-1}+\mathbf{A}_\infty \end{bmatrix} \mathbf{b} \end{aligned} \tag{6.63}$$

This *lemma* is proved by showing that all elements of $\mathbf{X}$ in equation-(6.63) are zero. In equation-(6.64), taking $[\mathbf{I}-\mathbf{A}]^{-1}$ common gives

$$\mathbf{X} = \mathbf{c}^T \begin{bmatrix} [\mathbf{A}-\mathbf{I}]+[\mathbf{I}-\mathbf{A}] \\ [\mathbf{A}^2-\mathbf{I}]+[\mathbf{I}+\mathbf{A}][\mathbf{I}-\mathbf{A}] \\ [\mathbf{A}^3-\mathbf{I}]+[\mathbf{I}+\mathbf{A}+\mathbf{A}^2][\mathbf{I}-\mathbf{A}] \\ \vdots \\ -\mathbf{I}+\mathbf{A}_\infty[\mathbf{I}-\mathbf{A}] \end{bmatrix} [\mathbf{I}-\mathbf{A}]^{-1}\mathbf{b} \tag{6.64}$$

It is clear that all elements inside the matrix in the square brackets in equation-(6.64) are zero, thus proving the *lemma*. For the last row in the above equation, the equality $\mathbf{A}_\infty^{-1} = \mathbf{I}-\mathbf{A}$ from equation-(6.24) has been used to prove this *lemma*. □

Corollary 6.1 *With an increase in frequency, the robustness bound for the Markov-Laguerre based MPC is influenced more by the controller parameters than the model.*

Proof: With an increase in the frequency: (a) influence of the model reduces if the magnitude spectrum of the model drops at the higher frequencies which usually is the case; and (b) the influence of Δ which comes from the controller in equation-(6.60) increases with increasing frequency.

If the noise model as shown by the equation-(6.40) is considered, then from equation-(6.56), it follows that the designed closed-loop can be expressed as

$$\Delta u(t) = -v_1 y(t) - v_2 y(t-1) - \mathbf{v}_3^T \mathbf{x}(t) - v_4 \Delta u(t-1) - v_5 u^f(t-1) \tag{6.65}$$

where $\mathbf{w} = 0$ for simplicity and where

$$\begin{aligned}
v_1 &= \mathbf{h}_1 \mathbf{S}^T \boldsymbol{\Gamma}_y \boldsymbol{\Xi} \\
v_2 &= \mathbf{h}_1 \mathbf{S}^T \boldsymbol{\Gamma}_y \boldsymbol{\Psi} \\
\mathbf{v}_3 &= \mathbf{h}_1 \mathbf{S}^T \boldsymbol{\Gamma}_y \boldsymbol{\Pi} \\
v_4 &= \mathbf{h}_1 \mathbf{S}^T \boldsymbol{\Gamma}_y \boldsymbol{\Omega} \\
v_5 &= \mathbf{h}_1 \mathbf{S}^T \boldsymbol{\Gamma}_y \boldsymbol{\Phi}
\end{aligned} \tag{6.66}$$

and $\mathbf{h}_1$ is the first row of $\mathbf{H}$.

As in the case of equation-(6.58), the designed closed-loop for the Markov-Laguerre based MPC with a noise model can be expressed by using equations-(6.4), (6.46), (6.47) and (6.65) to yield:

$$\begin{aligned}
&\left[\mathbf{x}(t+1), y(t+1), y^f(t+1), \Delta y(k+1), u^f(t+1), \Delta u(t+1)\right]^T = \\
&\underbrace{\begin{bmatrix}
\mathbf{A} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times 1} & \mathbf{b} & \mathbf{0}_{n\times 1} \\
\mathbf{c}^T\mathbf{A} & 1 & c & -d & \mathbf{c}^T\mathbf{b} & 0 \\
\mathbf{c}^T\mathbf{A} & 0 & 0 & 0 & \mathbf{c}^T\mathbf{b} & 0 \\
\mathbf{c}^T\mathbf{A} & 0 & c & -d & \mathbf{c}^T\mathbf{b} & 0 \\
-v_1\mathbf{c}^T\mathbf{A} - v_3\mathbf{A} & -v_1 - v_2 & -v_1 c & v_1 d & -v_1\mathbf{c}^T\mathbf{b} - \mathbf{v}_3\mathbf{b} - v_5 - c & -v_4 + d \\
-v_1\mathbf{c}^T\mathbf{A} - v_3\mathbf{A} & -v_1 - v_2 & -v_1 c & v_1 d & -v_1\mathbf{c}^T\mathbf{b} - \mathbf{v}_3\mathbf{b} - v_5 & -v_4
\end{bmatrix}}_{\mathbf{A}_c} \\
&\left[\mathbf{x}(t), y(t), y^f(t), \Delta y(t), u^f(t), \Delta u(t)\right]^T
\end{aligned} \tag{6.67}$$

The closed-loop system represented by equation-(6.67) is stable if the eigenvalues of $\mathbf{A}_c$ are inside the unit circle. When $C(q) = D(q) = 1$ *i.e.* when no noise model is considered then the above equation-(6.67) reduces to equation-(6.58).

In order to determine the robustness bound $|1/M(z)|$, equation-(6.65) is expressed as:

$$\Delta u(z) = -\left\{ \left(v_1 + \frac{v_2}{z}\right) y(z) + \mathbf{v}_3 \mathbf{x}(z) + \frac{1}{z}\left(v_4 + v_5 \frac{D(z)}{C(z)}\right) \Delta u(z) \right\} \tag{6.68}$$

which gives

$$
\begin{aligned}
\left|\frac{1}{M(z)}\right| &= \left|\frac{\tilde{d}(z)}{u(z)}\right| \\
&= \left|\frac{\Delta\left[1 + v_3[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}\frac{D(z)}{C(z)} + \frac{1}{z}\left(v_4 + v_5\frac{D(z)}{C(z)}\right)\right] + (v_1 + \frac{v_2}{z})\overbrace{\mathbf{c}^T[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}}^{\hat{G}(z)}}{-(v_1 + \frac{v_2}{z})}\right| \\
&= \left|-\left\{\frac{\Delta\left[1 + v_3[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}\frac{D(z)}{C(z)} + \frac{1}{z}\left(v_4 + v_5\frac{D(z)}{C(z)}\right)\right]}{(v_1 + \frac{v_2}{z})} + \hat{G}(z)\right\}\right| \qquad (6.69)
\end{aligned}
$$

after some rearrangement and substituting $\mathbf{x}(z) = [z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}u^f(z)$ and $y(z) = \mathbf{c}^T[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}u(z) + \tilde{d}(z)$ in equation-(6.68); where $\hat{d}(z) = \mathcal{Z}[\tilde{d}(t)]$. At $\omega = 0$, equation-(6.69) reduces to $|1/M(\omega)| = |\hat{G}(\omega)|$ because: $z = 1$ and $\Delta = 0$. From equation-(6.69) it is obvious that with an increase in frequency, $|1/M(\omega)|$ is influenced more by the controller terms because $|\hat{G}(\omega)|$ reduces and the influence of Δ increases.

6.7 Simulation Results

This section shows the use of the small gain theorem to evaluate the effect of robustness of different tuning parameters for Markov-Laguerre based MPC. The method used to evaluate the robustness of these tuning parameters is the same as presented in Chapter 5 for the GPC. Through simulations it is shown that the behavior of different tuning parameters of Markov-Laguerre MPC and GPC are similar.

The simulation results presented in this section are based on a noise-free third order overdamped plant: $G(s) = e^{-4s}/(s + 1)(3s + 1)(5s + 1)$ whose discrete equivalent for a sampling time $T_s = 1$ is:

$$
G(q) = \frac{[0.0077q^{-1} + 0.0212q^{-1} + 0.0036q^{-3}]q^{-4}}{1 - 1.9031q^{-1} + 1.1514q^{-2} - 0.2158q^{-3}} \qquad (6.70)
$$

The above mentioned plant is approximated by a 4^{th} order Markov and 2^{nd} order Laguerre model whose dominant pole is assumed to be at $a = 0.8434$. For this specified model structure, the LS identification method results in the following model for a square wave excitation:

$$
\hat{G}(q) = [0.0181, 0.0002, 0.0002, -0.0923, 0.2381, 0.0872][q^{-1}, \ldots, q^{-4}, L_1(a, q), L_2(a, q)]^T \qquad (6.71)
$$

Note 6.2 *The argument a in equation-(6.71) signifies that L_k is a function of a.*

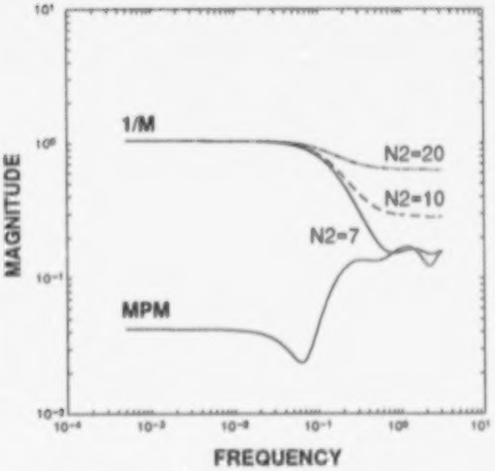


Figure 6.3: Effect of N_2 on the robustness margin.

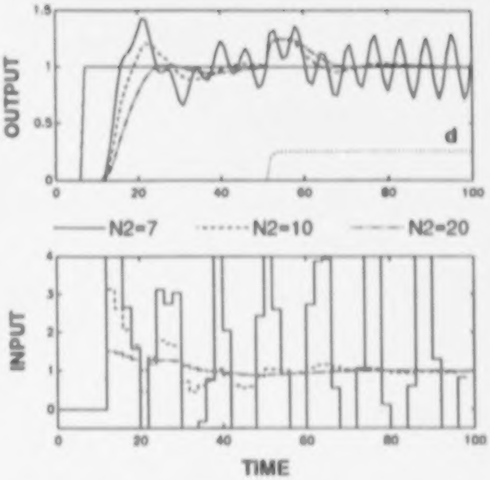


Figure 6.4: Effect of N_2 on the servo and regulatory response.

As discussed in Chapter 5, the mismatch between model and plant is deliberate, because we want to examine the stability of the Markov-Laguerre based MPC in the presence of MPM. subsections 6.7.1 to 6.7.4 discuss the effect of different tuning parameters on the robustness of Markov-Laguerre based MPC. A comparison between GPC and Markov-Laguerre based MPC is studied in subsection 6.7.5. The effect of the noise model on the performance of the Markov-Laguerre MPC is illustrated in subsection 6.7.6. For the sake of clarity in understanding the effect of tuning parameters on the robustness and performance, noise-free simulations are considered in most of these case studies.

6.7.1 Effect of N_2

As in the case of GPC, the prediction horizon N_2 plays an important role in imparting robustness to the Markov-Laguerre based MPC. This is verified in Figure 6.3, which shows spectral plots of the robustness bound $|1/M(\omega)|$ for different values of N_2 superimposed on magnitude spectrum of the uncertainty. The values of N_2 examined are 7, 10 and 20, with the other tuning parameters kept constant at $N_1 = 1$, $NU = 1$, $\lambda = 0$ and $\gamma_{\infty} = 0$. The spectral plots in Figure 6.3 shows that $N_2 > 7$ is required for this example (system (6.70) and (6.71)) in order to achieve robustness with respect to modelling errors. For $N_2 \leq 7$ the SGT stability criterion is violated at the higher frequencies which gives rise to instability as can be seen in Figure 6.4. Figures 6.3 and 6.4 show that with the increase in N_2 the controller becomes more detuned, consequently, robustness of the controller improves,

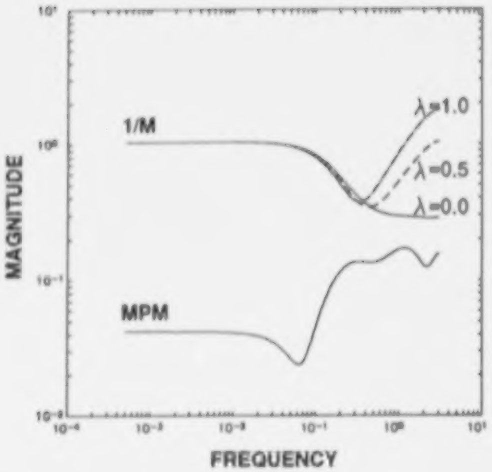


Figure 6.5: Effect of λ on the robustness margin.

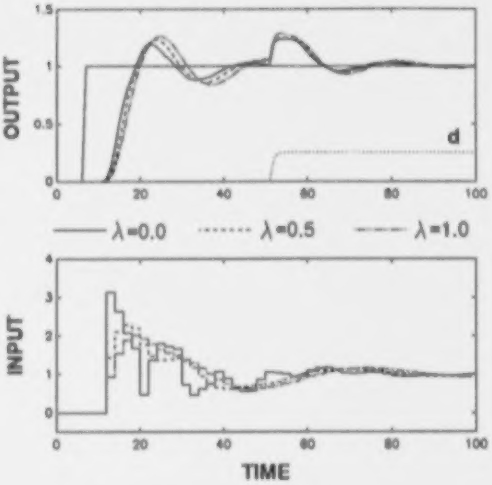


Figure 6.6: Effect of λ on the servo and regulatory response.

overshoot reduces and overall performance becomes more sluggish.

The choice of N_2 is also important to achieve a nominal stability, for instance, design of this controller for the model given by equation-(6.71) is stable only if $N_2 > 5$. Elshafei *et al.* [2] showed that irrespective of the choice of tuning parameters, the nominal design of Laguerre based MPC is always stable, but this observation does not hold for the Markov-Laguerre based MPC.

6.7.2 Effect of λ

It is known that the move supression factor λ detunes the controller performance hence the smallest practical value of λ should be used. Since λ restrains the controller movements, it is expected that it will also improve the robustness margin as observed in Figures 6.5 and 6.6 for the λ values of 0, 0.5 and 1.0 with other tuning parameters constant at $N_1 = 1$, $N_2 = 10$, $NU = 1$ and $\gamma_\infty = 0$. The influence of λ on the nature of robustness margin is however different from N_2. Figure 6.5 shows that λ produces a 'minimum' in $|1/M(\omega)|$ (similar to GPC as observed in Chapter 5) unlike N_2, thus making the controller sensitive around the frequencies where the dip occurs.

In Figure 6.4 it was observed that the system response varied significantly with the change in N_2, whereas Figure 6.6 shows no such appreciable variation in the process performance with the change in λ values. Although the controller detunes with the increase in λ, it only increases the overshoot although it is marginal and it does not seem to affect

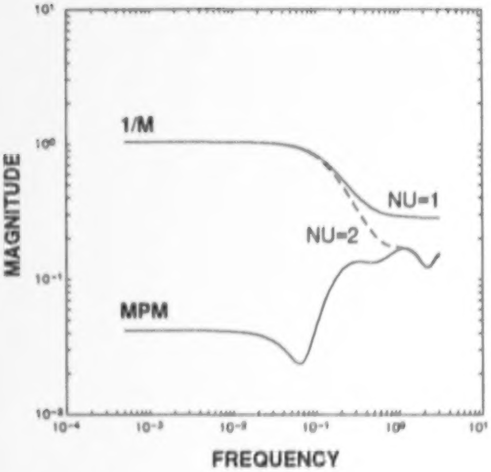


Figure 6.7: Effect of NU on the robustness margin.

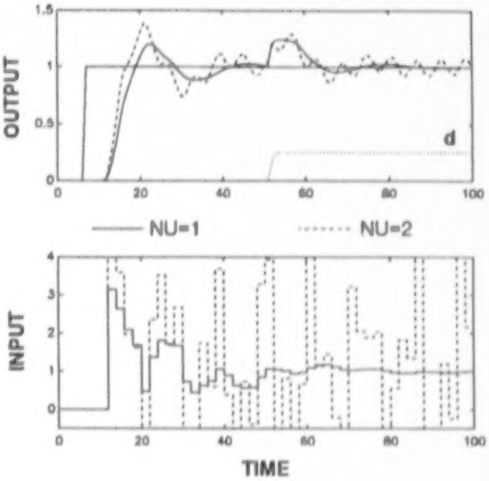


Figure 6.8: Effect of NU on the servo and regulatory response.

the regulatory response. Therefore as far as possible, the use of λ is not recommended as it produces a 'minimum' in $|1/M(\omega)|$ and does not seem to improve the process performance.

6.7.3 Effect of NU

The controller horizon NU adds to the aggressiveness in the control action, hence as in the case of GPC, the use of a large NU is expected to reduce the robustness margin. This is verified for NU values of 1 and 2 with the other tuning parameters kept constant at $N_1 = 1$, $N_2 = 10$, $\lambda = 0$ and $\gamma_\infty = 0$. Figure 6.7 shows that the robustness margin decreases as the value of NU is increased from 1 to 2. For $NU = 2$, the spectrum of $|1/M(\omega)|$ touches the MPM, consequently this makes the system unstable as depicted in Figure 6.8.

6.7.4 Effect of γ_∞

The steady state weighting term γ_∞ gives approximately the same effect as a large prediction horizon, without actually using a large value for N_2. Consequently it is expected that the use of γ_∞ will also increase the robustness margin in a similar way as in the case of N_2. This is observed in Figure 6.9 which shows that as the value of γ_∞ increases from 0 to 1, the robustness margin increases in a similar way as in the case of N_2. In Figures 6.9 and 6.10 the γ_∞ values are examined by keeping $N_1 = 1$, $N_2 = 10$, $NU = 1$ and $\lambda = 0$ constant. Figure 6.10 shows that with the increase in γ_∞: (a) overshoot in the servo response reduces; and (b) regulatory behavior becomes more detuned.

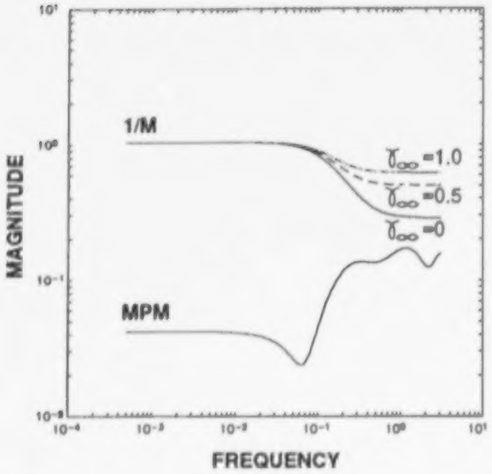


Figure 6.9: Effect of γ_∞ on the robustness margin.

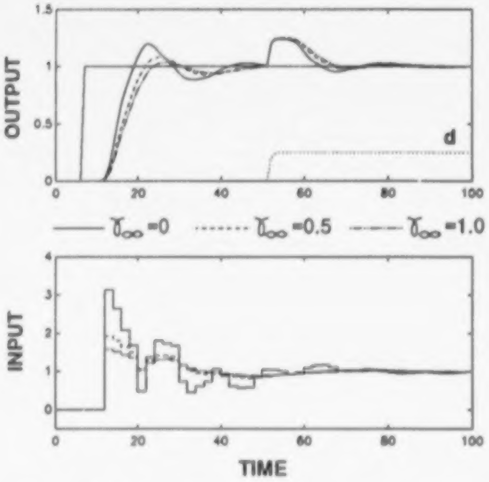


Figure 6.10: Effect of γ_∞ on the servo and regulatory response.

6.7.5 Comparison between GPC and Markov-Laguerrre MPC

This subsection compares the robustness and performance between GPC and Markov-Laguerre based MPC. A fair comparison between GPC and Markov-Laguerre MPC is possible only if: (a) both the ARX and Markov-Laguerre models have the same dynamics and almost the same number of model parameters; and (b) same set of tuning parameters be used for both the cases. It is difficult to satisfy the first condition because the structure of both these models are entirely different[4]. Nevertheless an approximate comparison between ARIMAX based GPC and Markov-Laguerre based MPC is possible if the dynamics of both these identified models are sufficiently close.

For the noise-free plant given by equation-(6.70), a first order ARX with 4 delays and 2 parameters in the numerator (3 parameters in all) is identified using a square wave excitation. For the same excitation a second order Laguerre model with 4 Markov coefficients (6 parameters in all) is estimated for this plant. The MPM associated with the identified ARX and Markov-Laguerre models in Figure 6.11 indicate that: (a) the Markov-Laguerrre model better represents the lower frequency range than the ARX model; and (b) at the higher frequencies, the ARX model is identified better. Nevertheless, $||\tilde{G}(\omega)||_\infty$ of both these models are approximately the same and they occur towards the higher frequen-

---

[4] In Section 6.2 it was shown that Markov-Laguerre model can be converted into an ARX model such that dynamics of both these models are identical. But the converted ARX model in such a case is of very high order. Further, GPC based on this ARX model would require a large value of N_2 to achieve stability and suitable performance.

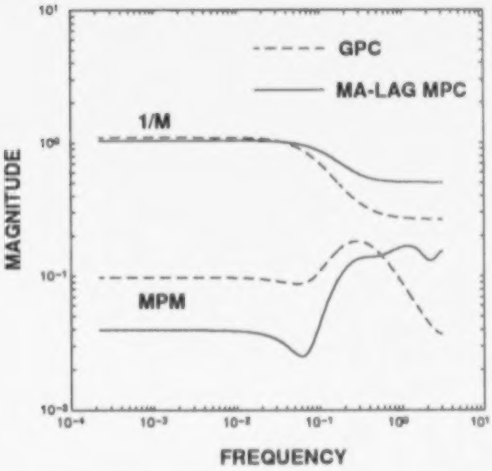


Figure 6.11: Robustness margins for GPC and Markov-Laguerre based MPC for a noise-free plant.

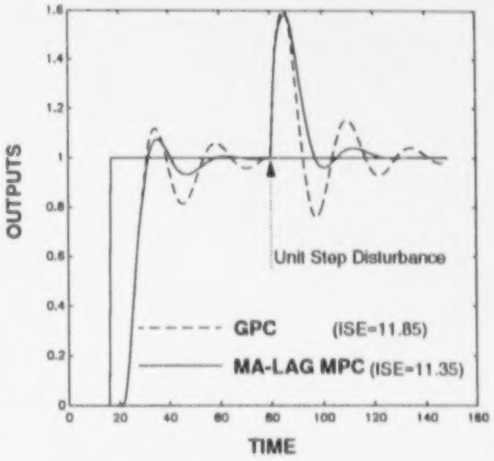


Figure 6.12: A comparative performance between GPC and Markov-Laguerre based MPC for a noise-free plant.

cies and also their time domain responses are similar (not shown). Hence the dynamics of both these models can be assumed to be similar and comparable which is sufficient for our analysis.

The robustness bounds of GPC and Markov-Laguerre MPC shown in Figure (6.11) are obtained by setting the tuning parameters constant at $N_1 = 1$, $N_2 = 10$, $NU = 1$, $\lambda = 0$, $\gamma_\infty = 0.5$ and in addition GPC uses $C_c(q) = 1 - 0.8q^{-1}$. Figure 6.11 indicates that the Markov-Laguerre MPC is more robust than GPC. This is because the MPC is designed without using any noise model and GPC uses a noise model of the type $C(q)/A(q)\Delta$. The corresponding performance in Figure 6.12 shows that the GPC response is more oscillatory (owing to smaller robustness margin) but it rejects the disturbance faster than the MPC (due to use of noise model in the GPC).

When the plant is corrupted by noise, and a stretched RBS excitation is used to identify the models, we see that the GPC becomes unstable whereas the Markov-Laguerre based MPC is stable as shown in Figure 6.13. GPC becomes unstable because it violates the small gain stability criteria due to a large MPM as shown in Figure 6.13. The ARX model required by the GPC is poorly identified because of high level of measurement noise ($\text{SNR} \approx 1.1$) and the excitation is improper for identification of the ARX model[5]. The instability can be contained through the input constraints by using a constrained controller as shown

---

[5] A combination of measurement noise and high frequency input leads to poor estimation of the ARX model, however in such a case the use of a low pass filter improves quality of the estimated model.

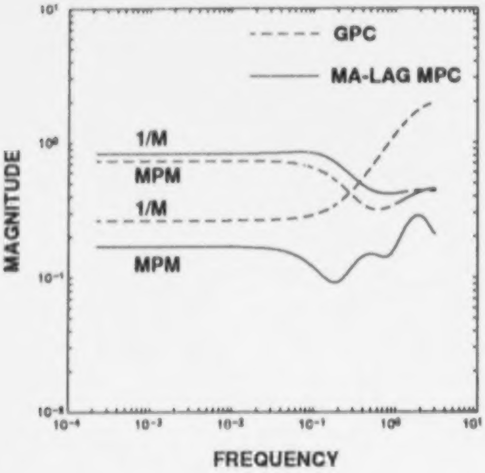


Figure 6.13: Robustness margins for GPC and Markov-Laguerre based MPC for a noisy plant.

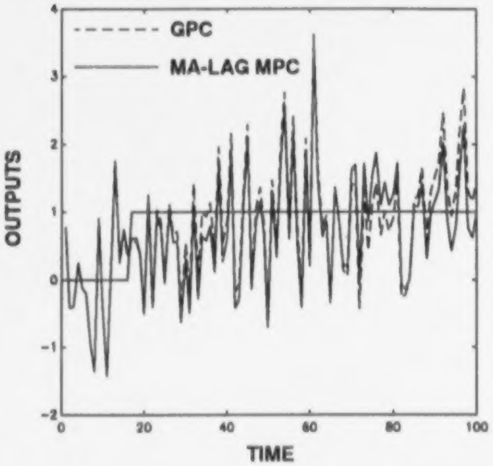


Figure 6.14: A comparative performance between GPC and Markov-Laguerre based MPC for a noisy plant.

in Figures 6.15 and 6.16. But in Figure 6.16 it is seen that the controller corresponding to the MPC hits the constraints more often.

For the same level of noise *i.e.* SNR$\approx$1.1, if a square wave excitation is used and if the number of Laguerre parameters be increased to 3 from 2, then in Figure 6.18 it is seen that the Markov-Laguerre based MPC becomes unstable whereas the GPC is stable. The Figure 6.17 robustness plot in shows violation of the stability criteria for the MPC due to large MPM at a certain frequency range for the Markov-Laguerre model. The square wave excitation gives a relatively better identification for the ARX model (due to more emphasis at the lower frequency range) whereas it create problems for the Markov-Laguerre model at the higher frequencies. These examples show that GPC or Markov-Laguerre MPC can become unstable if the corresponding MPM is large. The choice of identification design criteria is important and must be carefully selected depending on the model to be identified.

6.7.6 Effect of noise model on the performance of Markov-Laguerre MPC

The effect of including a noise model in the formulation of Markov-Laguerre based MPC is discussed in this subsection. Figures 6.19 and 6.20 compare the robustness and performance of Markov-Laguerre based MPCs that are formulated with and without the noise model. As in the case of GPC, the use of a noise model in the Markov-Laguerre MPC helps to achieve faster rejection of disturbance as shown in Figure 6.20. The use of a noise model

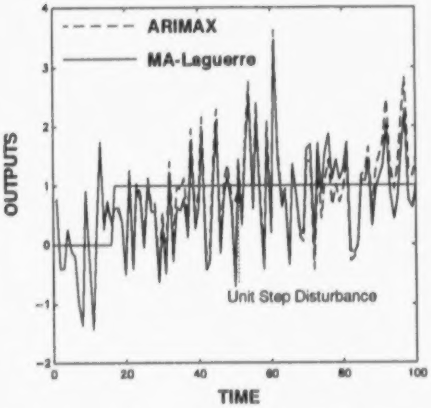


Figure 6.15: Performance of constrained GPC and Markov-Laguerre based MPC for a noisy plant.

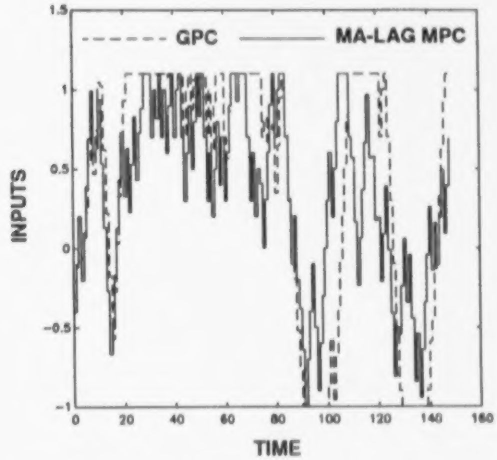


Figure 6.16: Input constraint stabilizes the system but the input hits the saturation limits more often.

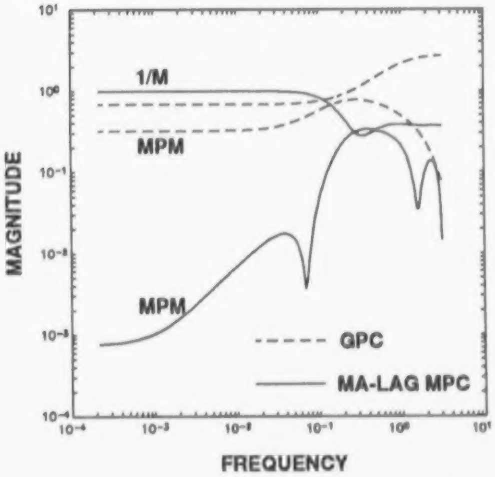


Figure 6.17: Robustness margins for GPC and Markov-Laguerre based MPC for a noisy plant.

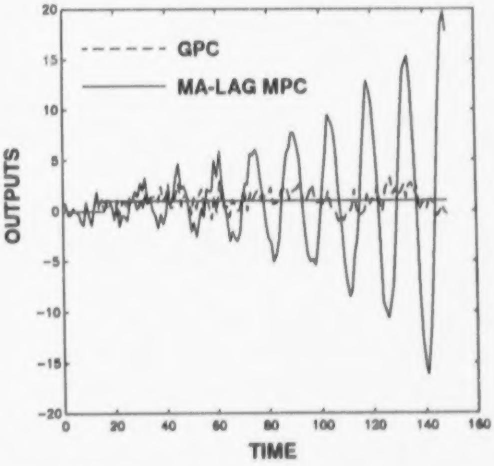


Figure 6.18: A comparative performance between GPC and Markov-Laguerre based MPC for a noisy plant.

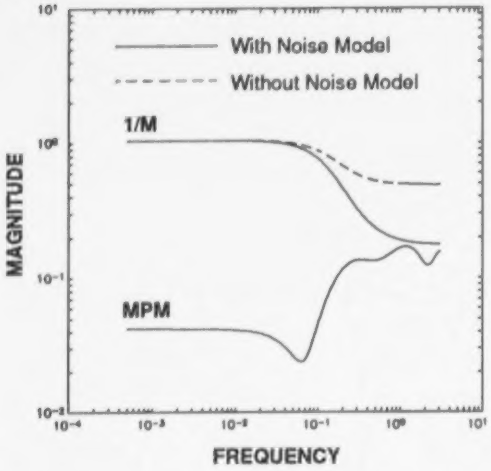


Figure 6.19: Robustness margins for Markov-Laguerre based MPCs with and without the noise model.

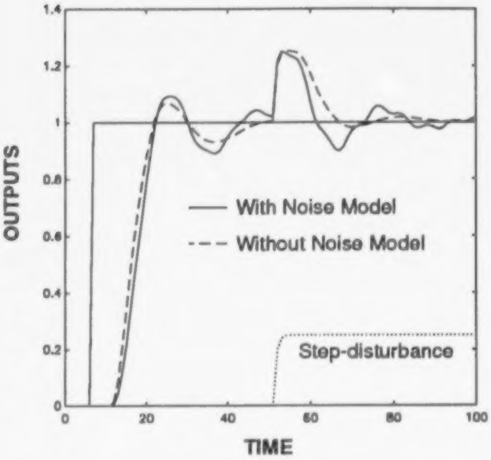


Figure 6.20: Regulatory behavior of the Markov-Laguerre based MPCs with and without the noise model.

also leads to a less robust controller design as depicted in Figure 6.19. For this particular example, $(1 - 0.25q^{-1})/((1 - 0.8q^{-1})\Delta)$ is selected as the noise model and this model is to be treated as an additional tuning parameter for the MPC to obtain faster disturbance rejecton. Similarly a comaprison between GPC and MPC with the noise model is made in Figures 6.21 and 6.22. With an appropriate selection of noise model for the MPC, these figures show that: (a) its robustness bound can be made similar to GPC; and (b) it can reject disturbance as well as GPC.

6.8 Conclusions

The main contributions of this chapter are as follows:

- A Markov-Laguerre model is used to formulate a model predictive controller (MPC) by first converting the Markov-Laguerre model into a state-space form. The objective function of this MPC is the same as GPC. Formulation of a Markov-Laguerre MPC with and without employing a noise model is considered.
- For MPC without a noise model: (a) a steady state weighting is incorporated in its formulation to enhance stability and performance of the controller; and (b) input/output constraints are incorporated and this problem is solved using quadratic programming.

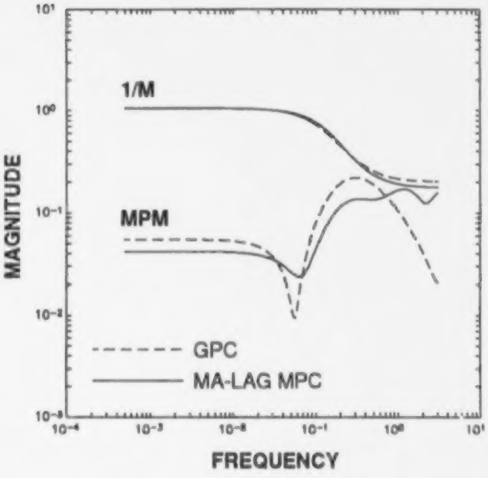


Figure 6.21: Robustness margins for GPC and Markov-Laguerre based MPC with a noise model.

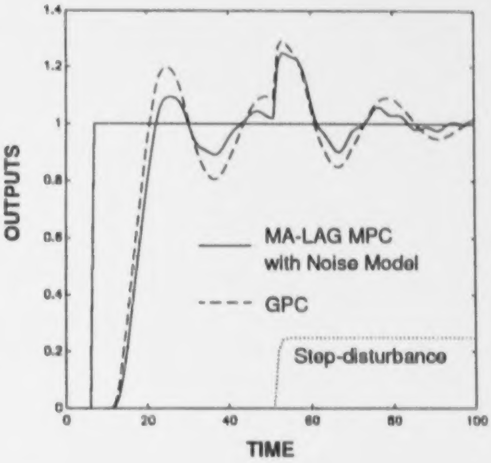


Figure 6.22: Performance of GPC and Markov-Laguerre based MPC with a noise model.

- A first order structured noise model (*i.e.* a combination of AR and MA models) is combined with the unstructured Markov-Laguerre model to formulate an unconstrained MPC. The noise model in the MPC is used as a tuning parameter to achieve faster rejection of the disturbances. The usual disturbance rejection property of MPC without the noise model is like DMC and slower than GPC, however this situation is remedied by including a suitable noise model in the formulation of MPC.

- The small gain theorem is used to evaluate the robustness of Markov-Laguerre model based MPC in the presence of modelling errors. Before evaluating the robustness, stability of the nominal design is checked by ascertaining if the eigenvalues of $\mathbf{A}_c$ given by equations-(6.58) or (6.67) are inside the unit circle. The robustness of different tuning parameters of MPC such as N_2, NU, λ and γ_∞ are examined through simulations and it is shown that these parameters affect the robustness margin $|1/M(\omega)|$ in a way similar to GPC. Inclusion of a noise model in the formulation of MPC can reduce its robustness margin.

- As in the case of GPC, the robustness bound $|1/M(\omega)|$ is a summation of the model $\hat{G}(\omega)$ and terms associated with the controller. At $\omega = 0$, $|1/M(\omega)|$ is equal to the steady state gain of the model and the robustness bound $|1/M(\omega)|$ is influenced more by the controller at the higher frequencies.

Bibliography

[1] C.C. Zervos and G.A. Dumont. "Deterministic Adaptive Control based on Laguerre Series Representation". *Int. J of Control*, 48(6):2333 – 2359, 1988.

[2] A. Elshafei, G.A. Dumont, and A. Elnaggar. "Adaptive GPC Based on Laguerre-filters Modelling". *Automatica*, 30(12):1913 – 1920, 1994.

[3] G.A. Dumont, A. Elnaggar, and A. Elshafei. "Adaptive Predictive Control of Systems with Time-Varying Time Delay". *Int. J. of Adaptive Control and Signal Processing*, 7(6):91 – 101, 1993.

[4] G.Y. Dumont, C.C. Zevros, and G.L. Pageau. "Laguerre-based Adaptive Control of pH in an Industrial Bleach Plant Extraction Stage". *Automatica*, 26(4):781 – 787, 1990.

[5] B.J. Allison, G.A. Dumont, L.H. Novak, and W.J. Cheetam. "Adaptive-Predictive Control of Kamyr Digester Chip level". *AIChEJ*, 36(7):1075 – 1086, 1990.

[6] C.K. Finn, B. Wahlberg, and B.E. Ydstie. "Constrained Predictive Control Using Orthonormal Expansions". *AIChEJ*, 39(11):1810 – 1826, 1993.

[7] K.Y. Kwok and S.L. Shah. "Long-Range Predictive Control with a Terminal Matching Condition". *Chem. Eng. Sci.*, 49(9):1287 – 1300, 1994.

[8] K. Kwok. *Long-Range Adaptive Predictive Control: Extensions and Applications.* Ph.D. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1992.

[9] H. Michalska and D.Q. Mayne. "Robust Receding Horizon Control of Constrained Nonlinear Systems". *IEEE Trans. on AC*, 38(11):1623–1633, 1993.

[10] J.B. Rawlings and K.R. Muske. "The stability of Constrained Receding Horizon Control". *IEEE Trans. on AC*, 38(10):1512–1520, 1993.

[11] M. Saudagar. *Unified Predictive Control.* PhD. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1995.

[12] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. "Generalized Predictive Control-Part I and II". *Automatica*, 23:137–160, 1987.

[13] L. Ljung. *System Identification-Theory For The User.* Prentice Hall, New Jersey, USA, 1987.

[14] G. Zames. "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems. Part-I: Conditions Using Concepts of Loop Gain, Conicity and Positivity". *IEEE Trans. on AC*, 11(2):228–238, 1966.

[15] P. Banerjee and S.L. Shah. "The Role of Signal Processing Methods in the Robust Design of Predictive Control". *Automatica*, 31(5):681 – 695, 1995.

[16] K. Qi. *Dual Model Based Robust Predictive Controller Design.* PhD. Thesis (under preparation), Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1996.

[17] J. Lunze. *Robust Multivariable Feedback Control.* Prentice Hall, New Jersey, USA, 1988.

Chapter 7

Control Relevant Identification for GPC

Different single-step and iterative control relevant identification methods are reviewed and evaluated by applying them to linear SISO GPC. The goal of control relevant identification is to enhance the controller performance. This is achieved by minimizing the controller and identification objectives in a synergistic manner.

7.1 Introduction

Chapters 2 to 4 of this thesis were concerned with the estimation of model from open-loop plant data. The use of such estimated models to design certainty equivalence based predictive controllers was subsequently described in Chapters 5 and 6. The main focus of Chapters 5 and 6 remained on the design of predictive controllers that are robust in the presence of model-plant mismatch (MPM).

An important issue related to robust stability is performance. In controller design one always endeavors to attain a good closed loop performance, because that is what ultimately matters. Since MPM is inevitable in all realistic cases, ***the pursuit for a good or desirable performance is feasible only when the system is robustly stable.***

Furthermore, the issue of robustness and performance is considered to be a problem of trade-off. For example, in Chapters 5 and 6 it was seen that the performance degraded as the robustness margin increased. On the other hand, in some cases it was also noticed in Chapters 5 and 6 that the system response became oscillatory (*i.e.* again a degradation in performance) due to lack of sufficient robustness margin at certain frequencies. Therefore

---

[1] A version of this chapter has been presented as: 'A Robustness Perspective on Control Relevant Identification for Long Range Predictive Control', by P.Banerjee and S.L.Shah at the 1995 AIChE Annual Meeting, Miami, Florida, USA.

a question that naturally arises is: *what robustness margin gives an acceptable performance ?* or stated differently *can system performance be predicted by specifying the robustness margin ?* Unfortunately, this is a difficult question to address and the system performance can seldom be quantitatively predicted by examining the frequency domain based robustness margins. Nevertheless, the system performance can be qualitatively assessed by observing the nature of robustness margin and this observation was utilized in Chapter 5 in the context of tuning GPC parameters using an optimization technique. This motivates one to look for alternate ways of enhancing the performance which is the subject of discussion in this chapter.

The quest for performance improvement in certainty eqivalence controllers has led researchers to have a fresh look at identification issues. Therefore a pertinent question that arises in this context is: *what model will result in good controller performance ?* or alternately *can a model be identified in such a way that it will yield desirable controller performance ?* In an excellent tutorial review on identification (ID) for control, Gevers [1] noted that: "*When a reduced complexity model is identified with the purpose of designing a robust controller, the model is just a vehicle for the computation of a controller*" and an important overall objective should be: "*the global control performance criteria must determine the identification criteria*". Infact several researchers [2, 3, 4, 5] have recognized this need for a suitable model ID scheme for control, some of which have been reviewed in this chapter from the point of view of improving the performance of GPC.

The importance of system ID to estimate models for the design of a certainty equivalence controller was recognized as early as the 60s when the certainty equivalence controller was first proposed [1][2]. Since then the major effort in system ID went into the issues related to the development of a good open-loop model, but the effect of undermodelling on the controller performance was largely left out.

Later, the need for accounting for modelling error in controller design gave rise to the field of robust controller design in the 1980s[3] [1]. These robust design procedures are usually frequency domain based and require some *a priori* knowledge of the MPM[4] without accounting for this MPM is estimated. Thus it was seen that although system ID and robust controller design emerged to facilitate better and reliable design for the certainty eqivalence controllers, they were developed as mutually exclusive entities by themselves. For example, the lack of accord between system ID and robust controller design is more apparent in the context of adaptive control; where: (a) it is difficult to perceive if the identified model will keep the system stable; and (b) frequency domain based robust controller design cannot be used to ascertain the stability of a system[5].

---

[2] R.E.Kalman first published state-space model based controller design in 1960 as: "Contributions to the theory of optimal control", *Bol. Soc. Mathematica Mexicana*, pp 102-119, 1960 [1].

[3] The work by G.Zames, 'Feedback and Optimal Sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses.', *IEEE Trans. on Automatic Control*, AC-26, pp301-320, 1981, first marked the beginning of robust controller design [1].

[4] As seen in Chapters 5 and 6.

The need for coherency between system identification and controller design was realized as early as in 1960 by Fel'dbaum in the context of adaptive (stochastic) control for certain problems in communications where the process was poorly known. In order to handle these problems, Fel'dbaum introduced the concept of *dual control* that pursues the dual goal of identification and control simultaneously [6, 7]. The key strategy in dual control is to sacrifice short-term control performance for good overall long-term regulation, by 'rich' initial excitation to obtain good information on the process [8]. Unfortunately, dual control is impractical in most cases because of the nonlinear dependence of the control action on the parameter estimates [8]. ***Nevertheless, the estimation of a model that is commensurate with the control objectives has been the cornerstone of dual control philosophy and this recently has been the guiding principle for doing system identification and control in a synergistic manner.***

Fel'dbaum's work inspired Åström and Wittenmark to use a joint identification and control strategy to solve an optimal control problem with no MPM by employing dynamic programming [9]. Their work in essence showed that a joint strategy of identification and control: (a) is feasible for simple cases with no MPM; and (b) there are often multiple local minima in the cost function, and the actual global minima may be difficult to find [1, 8, 10].

The interplay between identification and control has received much recent attention in the literature under headings such as: ***control-relevant identification, iterative identification methods for control, control directed identification... etc.*** These recent investigations have focussed on the subject of ***how to obtain models that are relevant and appropriate for the control objectives and with a view to improve the controller performance.***

Interest in control relevant identification resurfaced in the early 1990s. Different researchers have provided different ways to integrate the dual objective of identification and control. The underlying principle and the common thread in all these identification methods are presented in Section 7.2.

The control relevant identification methods have been classified in this chapter as: one-step and multiple-steps or iterative designs. Under the one-step method, the open-loop model is used to design a suitable control relevant filter, which is used to subsequently obtain a control-relevant model. For the iterative design method, the identified control-relevant model is updated quasi-adaptively[6] until there is no improvement in closed-loop performance.

An identification scheme compatible with the long range predictive control (LRPC) strategy was proposed by Shook and Shah [8, 2] (in 1992) and it was labelled as the long range predictive identification (LRPI). This technique is reviewed briefly in Section 7.3.

---

[5] A small step in bridging the gap between robust design and adaptive control has been made by applying signal processing methods (in Chapter 2) to the method presented in Section 5.16 in this thesis.

[6] *i.e.* using a batch of closed-loop data.

The LRPI is a one-step scheme and it has been shown to work well for real-time adaptive control [8][7] as well.

Another one-step control relevant ID scheme was proposed by Rivera *et al.* in 1990; which has been successfully tested on industrial plants [11, 3, 12]. Rivera *et al.*'s control relevant identification scheme is more generic and has been applied to an LRPC[8] scheme. Furthermore, this control-relevant ID approach provides an alternative viewpoint when compared with LRPI and therefore this method is also discussed briefly in Section 7.4.

Zang *et al.* in 1991 and Schrama *et al.* in 1992 showed the need for iteration in upgrading the model and controller to achieve a better performance [1, 4, 13, 14]. They devised their iterative control relevant identification scheme for LQG based control. The principle behind these iterative controller design methods are similar, therefore only one of these approaches *i.e.* due to Zang *et al.* has been considered for review in Section 7.5. Zang *et al.* also proposed an (interesting) frequency weighting criteria in the objective function for the LQG controller; this idea has been applied to modify the GPC objective and is discussed in Section 7.6.3. The iterative implementation of the control-relevant ID scheme due to Zang *et al.* for GPC has also been discussed in this section. Specifically, the application of the control-relevant filters due to Zang *et al.* , Shook *et al.* and Rivera *et al.* to iteratively improve the performance of GPC are discussed in subsection 7.6.3 to 7.6.2. The contributions made in the preceeding sections are summarized in Section 7.7 as concluding remarks.

No matter how a model is obtained some amount of MPM is inevitable in all practical model-based controllers. There is therefore a need investigate *how the identified control relevant models shape the robustness margins for the LRPC* (GPC in this case). This work therefore also investigates how the measure of robustness, robustness margin and the shape of MPM are influenced by various *control-relevant identification* techniques.

7.2 Problem formulation

The entire issue of robust stability and performance of a certainty equivalence controller lies in the comparison between Figures 7.1 and 7.2. The closed-loop system represented by Figure 7.1 is defined as the *designed-loop* or the *designed-system*; similarly, the system shown in Figure 7.2 is defined as the *achieved-loop* or the *achieved-system*. A designed-system is based on the assumption that the model $\hat{G}(q)$ truly represents the plant dynamics, consequently its *designed* performance criteria $\hat{J}$ is given by:

---

[7] Although LRPI has been implemented for adaptive control, it cannot be considered to be a multiple-step method. A multiple-step or iterative version of LRPI is proposed in this chapter.

[8] specifically to *horizon predictive control* [12].

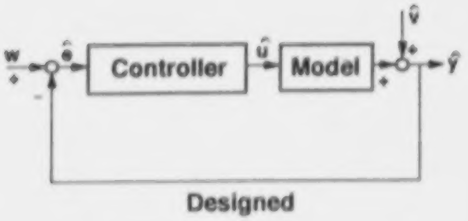


Figure 7.1: Designed closed-loop system.

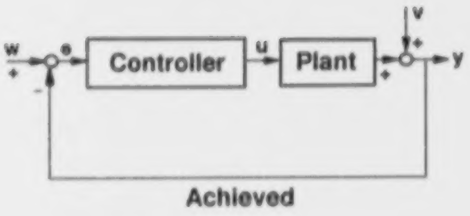


Figure 7.2: Achieved closed-loop system.

$$\hat{J} = \frac{1}{N}\sum_{i=1}^{N}(w(i) - \hat{y}(i))^2 + \lambda\sum_{i=1}^{N}\hat{u}^2(i) \tag{7.1}$$

Infact the LQG controller is designed to minimize the above stated performance criteria $\hat{J}$, when $N \to \infty$. When the designed controller is implemented on the actual plant as shown in Figure 7.2, it results in a closed loop system which is different from the designed case due to the presence of modelling errors. The performance of the realized or the achieved-loop system is given by:

$$J = \frac{1}{N}\sum_{i=1}^{N}(w(i) - y(i))^2 + \lambda\sum_{i=1}^{N}u(i) \tag{7.2}$$

which is usually poor in comparison to $\hat{J}$ (under the influence of MPM). For the achieved loop, the issues which take precedence are:

- *Robust Stability*
- *Robust Performance*[9]

Robust stability is concerned with the stability of the achieved loop in presence of MPM. The use of the small gain theorem to examine the robust stability of model predictive controllers were discussed in Chapters 5 and 6. ***Robust performance*** on the other hand involves bringing J and $\hat{J}$ close to each other [1]. Obviously:

Remark 7.1 *Robust stability is a necessary but not a sufficient condition for robust performance.*

Note 7.1 *The controller under question in this study is GPC whose objective function is different from that is given by equations-(7.1) or (7.2). Nevertheless, these objective functions truly reflect the performance of the designed and the achieved loop, hence they are shown first to facilitate proper understanding of the problem. The developments based on equations-(7.1) and (7.2) are subsequently applied to GPC.*

---

[9] Although this term appears to be self-contradictory, it is defined in the following paragraph.

The endeavor in achieving robust performance is to minimize J rather than focusing on minimizing $\hat{J}$, because the quantity that ultimately matters is J. For a moment if it is assumed that $\lambda = 0$, then J in equation-(7.2) can be rewritten as:

$$\begin{aligned} J &= \frac{1}{N}\sum_{i=1}^{N}\{w(i) - \hat{y}(i) + \hat{y}(i) - y(i)\}^2 \\ &= \underbrace{\frac{1}{N}\sum_{i=1}^{N}\{w(i) - \hat{y}(i)\}^2}_{\hat{J}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\{\hat{y}(i) - y(i)\}^2}_{J_{id}} + \underbrace{\frac{2}{N}\sum_{i=1}^{N}\{[w(i) - \hat{y}(i)][\hat{y}(i) - y(i)]\}}_{J_{cross}} \end{aligned} \tag{7.3}$$

Direct minimization of J in equation-(7.2) is the problem of dual control which is difficult to solve [8]. However equation-(7.3) shows that J can be split into the designed control objective $\hat{J}$, the identification objective J_{id} and the cross product terms designated as J_{cross}. It is difficult to handle the cross product terms J_{cross}, hence this term is neglected [8]. Therefore the problem of minimizing J reduces to a suboptimal problem and applying the triangular inequality[10] to equation-(7.3) gives:

$$\sqrt{J} \leq \sqrt{\hat{J}} + \sqrt{J_{id}} \tag{7.4}$$

Since a simultaneous minimization of $\hat{J}$ and J_{id} is difficult, hence the strategy followed for *all* control-relevant identification methods is:

Remark 7.2 *Minimize J by separately minimizing $\hat{J}$ through proper controller design and by minimizing J_{id} by using a suitable system ID method.*

One of the principal goals of control-relevant ID is to minimize J_{id}. For most control relevant ID methods (at least those discussed in this chapter), the least-squares (LS) method or the prediction-error method (PEM) is used to obtain a suitable control-relevant model from the filtered open-loop or closed-loop data; where an appropriate ***control-relevant filter*** is used to filter the process data for ID. An important step in the control-relevant ID is to obtain this control-relevant filter by equating J_{id} with the open-loop or the closed-loop LS objectives.

For a general case *i.e.* $\lambda \neq 0$ and for the performance criteria given by equations-(7.1) and (7.2), Gevers [1] showed that the achieved performance J is bounded between the two limits as:

---

[10] The triangular inequality is given by: $\|a + b\| \leq \|a\| + \|b\|$ where a, and b are real or complex numbers [15]. Similar to the triangular inequality is the Minkowski's inequality which is applicable in this case. The Minkowski inequality applies to finite sums and it is given by: $\{\sum_{i=1}^{N}|a(i) \pm b(i)|^p\}^{1/p} \leq \{\sum_{i=1}^{N}|a(i)|^p\}^{1/p} + \{\sum_{i=1}^{N}|b(i)|^p\}^{1/p}$ where a and b are real or complex numbers [15].

$$|\sqrt{\hat{J}} - \sqrt{J_{id}}| \leq \sqrt{J} \leq |\sqrt{\hat{J}} + \sqrt{J_{id}}|$$

where

$$J_{id} = \frac{1}{N}\left\{\sum_{i=1}^{N}[\hat{y}(i) - y(i)]^2 + \lambda\sum_{i=1}^{N}[u(i) - \hat{u}(i)]^2\right\} \tag{7.5}$$

The bounds given by equation-(7.5) are obtained by defining [1]:

$$\begin{aligned} \left\| \begin{matrix} w(i) - y(i) \\ \lambda u(i) \end{matrix} \right\|_2 &= \sqrt{\sum_{i=1}^{N}\left[\left(\frac{w(i) - y(i)}{\sqrt{N}}\right)^2 + \left(\frac{\lambda u(i)}{\sqrt{N}}\right)^2\right]} \\ \Rightarrow \qquad J &= \left\| \begin{matrix} w(i) - y(i) \\ \lambda u(i) \end{matrix} \right\|_2^2 \end{aligned} \tag{7.6}$$

and using the triagular inequality on:

$$\begin{pmatrix} w(i) - y(i) \\ \lambda u(i) \end{pmatrix} = \begin{pmatrix} w(i) - \hat{y}(i) \\ \lambda \hat{u}(i) \end{pmatrix} + \begin{pmatrix} \hat{y}(i) - y(i) \\ \lambda(u(i) - \hat{u}(i)) \end{pmatrix} \tag{7.7}$$

7.2.1 Performance criteria for GPC

This subsection presents different ways of expressing the performance measure of GPC. Traditionally, the GPC controller law at an instant t is defined as [16]:

$$J^{GPC}(t) = \sum_{i=N_1}^{N_2}[w(t+i) - \hat{y}(t+i|t)]^2 + \lambda\sum_{i=1}^{NU}[\Delta u(t+i-1)]^2 \tag{7.8}$$

For a specified setpoint trajectory, the designed GPC loop results in a set of input-output (I/O) data represented as $\hat{\mathcal{D}}(N) = \{\hat{u}(i), \hat{y}(i), i = 1 \dots N\}$, where N is the number of data points. Similarly for the achieved loop, the set of I/O data obtained is $\mathcal{D}(N) = \{u(i), y(i), i = 1 \dots N\}$.

Since the GPC is implemented using a receding horizon policy, the designed and the achieved performance of GPC can be respectively defined as (as in case of equations-(7.1) and (7.2)):

$$\begin{aligned} \hat{J}_{GPC} &= \frac{1}{N}\sum_{i=1}^{N}\{(w(i) - \hat{y}(i))^2 + \lambda\Delta\hat{u}^2(i)\} \\ J_{GPC} &= \frac{1}{N}\sum_{i=1}^{N}\{(w(i) - y(i))^2 + \lambda\Delta u^2(i)\} \end{aligned} \tag{7.9}$$

To be in confirmity with equation-(7.8), $\Delta\hat{u}$ and Δu have been used in equation-(7.9) instead of $\hat{u}$ or u as in equations-(7.1) or (7.2).

Note 7.2 *Note the difference between $J^{GPC}(t)$ in equation-(7.8) and J_{GPC} in equation-(7.9).*

If the prediction horizon N_1 to N_2 for the GPC is also considered at each time step $t, t = 1 \ldots N$, then the designed and the achieved objectives for the GPC can be expressed as:

$$\begin{aligned}
\hat{J}_{LR} &= \frac{1}{N-N_2}\sum_{t=1}^{N-N_2}\left\{\sum_{i=N_1}^{N_2}[w(t+i)-\hat{\hat{y}}(t+i|t)]^2+\lambda\sum_{i=1}^{NU}[\Delta\hat{\hat{u}}(t+i-1)]^2\right\} \\
J^{*}_{LR} &= \frac{1}{N-N_2}\sum_{t=1}^{N-N_2}\left\{\sum_{i=N_1}^{N_2}[w(t+i)-y^{*}(t+i|t)]^2+\lambda\sum_{i=1}^{NU}[\Delta u^{*}(t+i-1)]^2\right\}
\end{aligned} \tag{7.10}$$

where at each sampling instant t, the implemented controller action is given by $\hat{u}(t) = \hat{\hat{u}}(t)$ and $u(t) = u^{*}(t)$ according to the receding horizon strategy. Further, the subscript LR in the above equation denotes that the long-range prediction (LRP) is used for calculation of the objective at each sampling time t. Using equations-(5.8) to (5.12) in Chapter 5, the predictors $\hat{\hat{y}}(t+i|t)$ and $y^{*}(t+i|t)$ in equation-(7.10) can be written as:

$$\begin{aligned}
\hat{\hat{y}}(t+i|t) &= \mathcal{G}_i(q)\Delta\hat{\hat{u}}(t+i-1)+\tilde{G}_i(q)\Delta\hat{u}^{f}(t-1)+F_i(q)\hat{y}^{f}(t) \\
y^{*}(t+i|t) &= \mathcal{G}_i(q)\Delta u^{*}(t+i-1)+\tilde{G}_i(q)\Delta u^{f}(t-1)+F_i(q)y^{f}(t)
\end{aligned} \tag{7.11}$$

where $\hat{u}^{f}(t-1) = \hat{u}(t-1)/C_c(q)$, $u^{f}(t-1) = u(t-1)/C_c(q)$, $\hat{y}^{f}(t) = \hat{y}(t)/C_c(q)$ and $y^{f}(t) = u(t-1)/C_c(q)$. Another form of the achieved LRP performance objective can be:

$$J_{LR} = \frac{1}{N-N_2}\sum_{t=1}^{N-N_2}\left\{\sum_{i=N_1}^{N_2}[w(t+i)-y(t+i|t)]^2+\lambda\sum_{i=1}^{NU}[\Delta u(t+i-1)]^2\right\} \tag{7.12}$$

No matter how the performance objective is specified, using the triangular inequality it can be shown that the achieved performance objective is bounded below the corresponding controller and the ID objectives. In general it can be stated that:

$$|J| \leq |\hat{J}| + |J_{id}| \tag{7.13}$$

In this subsection, essentially two sets of performance specifications have been presented, *i.e.* equation-(7.9) and equation-(7.10) together with equation-(7.12). Out of these two sets of equations, the expressions in (7.9) truly reflect the performance measure, hence this equation has been used to evaluate the performance of GPC in the subsequent sections. Furthermore, equation-(7.12) has been used in a slightly modified form for the derivation of a control-relevant filter presented in the following section, nevertheless its performance has been assessed using equation-(7.9).

7.3 Control-relevant identification by Shook and Shah

A one-step control-relevant ID technique that is compatible with the GPC was proposed by Shook and Shah [2, 8] and they termed it as a long range predictive identification (LRPI). The basic philosophy of LRPI is to estimate a model that can best predict the future output of the process over a given prediction horizon as specified for the LRPC objective. Such an identification scheme is obtained by equating J_{id} for the GPC with the open-loop LS criteria J_{LS}^{OL}. J_{id} for the LRPI is obtained by adding and subtracting $y^*(t+i|t)$ as in equation-(7.3) for the following achieved performance specification (note the similarity of the following expression with equation-(7.12)):

$$J_{GPC}^{ach} = \frac{1}{N-N_2} \sum_{t=1}^{N-N_2} \frac{1}{N_p} \sum_{i=N_1}^{N_2} \{y(t+i) - w(t+i)\}^2 \tag{7.14}$$

to yield

$$J_{LRPI} = \frac{1}{N-N_2} \sum_{t=1}^{N-N_2} \left\{ \frac{1}{N_p} \sum_{i=N_1}^{N_2} [y(t+i) - y^*(t+i|t)]^2 \right\} \tag{7.15}$$

where $N_p = N_2 - N_1 + 1$. Substituting $y^*(t+i|t)$ from equation-(7.11) in equation-(7.15) gives:

$$\begin{aligned} J_{LRPI} \quad = \quad & \frac{1}{N-N_2} \sum_{t=1}^{N-N_2} \left\{ \frac{1}{N_p} \sum_{i=N_1}^{N_2} \left[y(t+i) - \left(\frac{F_i(q)y(t)}{C_c(q)} + \right.\right.\right. \\ & \left.\left.\left. \frac{E_i(q)\hat{B}^*(q)}{C_c(q)} \Delta u(t+i-1) \right) \right]^2 \right\} \end{aligned} \tag{7.16}$$

Note: refer Note-5.2 in Chapter 5, for the definition of the term $\hat{B}^*(q)$. Further rearranging the Diophantine equation-(5.7) (for $A(q) = \hat{A}(q)$, $P(q) = 1$) in Chapter 5, to:

$$1 = E_i(q)\Delta\hat{A}(q)/C_c(q) + q^{-i}F_i(q)/C_c(q) \tag{7.17}$$

and multiplying equation-(7.17) by $y(t+i)$ and substituting it in equation-(7.16) for $y(t+i)$ and also using $y(t+i) = B^*(q)u(t+i-1)/A(q)$ gives:

$$J_{LRPI} = \frac{1}{N - N_2} \sum_{t=1}^{N-N_2} \left\{ \frac{1}{N_p} \sum_{i=N_1}^{N_2} \left[\left(\frac{E_i(q)\Delta \hat{A}(q)}{C_c(q)} \right) \tilde{G}(q)u(t+i-1) \right]^2 \right\} \tag{7.18}$$

where $\tilde{G}(q) = G(q) - \hat{G}(q)$. The filter for LRPI is obtained by equating-(7.18) with the open-loop LS objective which is given by:

$$\begin{aligned} J_{LS}^{OL} &= \frac{1}{N} \sum_{i=1}^{N} \epsilon^2(i) \\ \text{where } \epsilon(i) &= y(i) - \hat{y}(i|i-1) \end{aligned} \tag{7.19}$$

In order to obtain control-relevant filter $L(q)$, $\epsilon(i)$ is multiplied with $L(q)$ to obtain filtered residual as: $\epsilon^f(i) = L(q)\epsilon(i)$. The model parameters θ are then obtained as $\theta = \arg\min J_{LS}^{OL}$. The expression for filter $L(q)$ is obtained by expressing $\epsilon^f(i)$ in terms of the MPM and the noise model. This expression is derived by considering the following model [17]:

$$\begin{aligned} y(t) &= \hat{G}(q)u(t) + \underbrace{\hat{H}(q)\xi(t)}_{v'(t)} \\ \Rightarrow \quad & y(t) = \hat{G}(q)u(t) + (\hat{H}(q) - 1)\xi(t) + \xi(t) \end{aligned} \tag{7.20}$$

where $\hat{H}(q)$ is the noise model and $\xi(t)$ is the white noise sequence. The first expression in equation-(7.20) can also be rearranged to: $\xi(t) = \hat{H}^{-1}(q)\ (y(t) - \hat{G}(q)u(t))$ and substituting it in the middle term in the second expression in equation-(7.20) gives the following (after taking one-step ahead prediction $\Rightarrow \mathcal{E}[\xi(t+1)] = 0$):

$$\hat{y}(t|t-1) = \hat{H}^{-1}(q)\hat{G}(q)u(t) + (1 - \hat{H}^{-1}(q))y(t) \tag{7.21}$$

Also the expression for the true plant is given by:

$$y(t) = G(q)u(t) + \underbrace{H(q)\xi(t)}_{v(t)} \tag{7.22}$$

where $G(q)$ and $H(q)$ are the true (and unknown) plant and the noise model. Therefore substituting the expressions for $y(t)$ from equation-(7.22), $\hat{y}(t|t-1)$ from equation-(7.21) into equation-(7.19), via $\epsilon^f(t)$ gives:

$$J_{LS}^{OL} = \frac{L(q)}{\hat{H}(q)} \left\{ \tilde{G}(q)u(t) + v(t) \right\} \tag{7.23}$$

The expresson for the filter $L(q)$ is obtained by expressing equations-(7.18) and (7.23) in the frequency domain (by using Parseval's theorem) and equating them. Therefore applying Parseval's theorem to equation-(7.18) as

$$J_{LRPI} = \frac{1}{N_p}\sum_{i=N_1}^{N_2}\frac{T_s}{2\pi}\int_{-\pi/T_s}^{\pi/T_s}\left|\frac{E_i(\omega)\Delta\hat{A}(\omega)}{C_c(\omega)}\right|^2 |\tilde{G}(\omega)|^2\Phi_{uu}(\omega)d\omega \tag{7.24}$$

(where T_s in the above equation is the sampling time and this term will be assumed to be one in all subsequent derivations for simplicity) and to equation-(7.23) as

$$J_{LS}^{OL} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left\{|\tilde{G}(\omega)|^2\Phi_{uu}(\omega)+\Phi_{vv}(\omega)\right\}\left|\frac{L(\omega)}{\hat{H}(\omega)}\right|^2 d\omega \tag{7.25}$$

and equating (7.24) and (7.25) and in equation-(7.25): (a) neglecting the term $\Phi_{vv}(\omega)$; and (b) using $\hat{H}(q) = C_c(q)/\Delta\hat{A}(q)$ gives the following expression for the LRPI filter $L(q)$:

$$|L(q)|^2 = \frac{1}{N_p}\sum_{i=N_1}^{N_2}|E_i(q)|^2 \tag{7.26}$$

The above LRPI filter is designated as the $L_s(q)$-filter in this chapter. Shook *et al.* suggested a polynomial deconvolution based spectral factorization method to solve equation-(7.26). Although this method of spectral factorization is approximate, nevertheless it gives a good estimate of $L_s(q)$. A more accurate spectral factorization is obtained from the formulation of the Riccatti equation. A MATLAB© based function 'sfl.m' can be used for this purpose.

Shook [8] noted the following features for the L_s-filter as: (a) it is a strong function of the prediction horizon N_2 and it is weakly dependent on $\hat{A}$; (b) typically L_s is a low-pass filter, but for a lightly damped system with a slow sampling rate L_s tends to become a high-pass filter. The L_s filter is used in conjunction with the Δ/C_c as shown in Figure 7.3; and therefore this filter gives the effect of band-pass filtering.

For the plant-model system described by equations-(5.51) and (5.52) in Chapter 5, the robustness bounds for the GPC corresponding to the models obtained using the LS and the LRPI methods are compared in Figure 7.4.

Note 7.3 *All simulation results presented in this chapter are based on the plant-model system given by equations-(5.51) and (5.52) in Chapter 5.*

Figure 7.4 shows that the nature of MPM associated with the LS based model is significantly different from the LRPI based model. The LS based model shows a good match at the higher frequencies whereas the model fit is poor in the mid frequencies. Consequently for this case, the robustness margin is poor at those frequencies where the MPM is significant. On the other hand, MPM associated with the LRPI based model shows a flatter spectrum. This is obtained by compromising the model fit at the lower frequencies and thereby reducing the

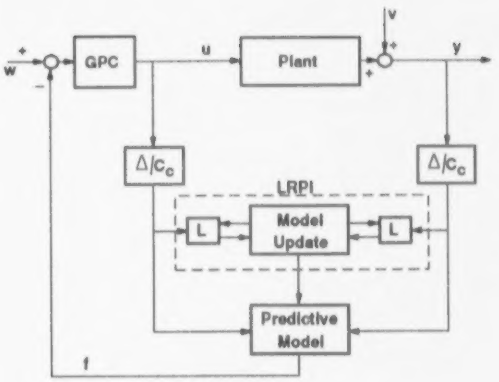


Figure 7.3: Implementation of LRPI.

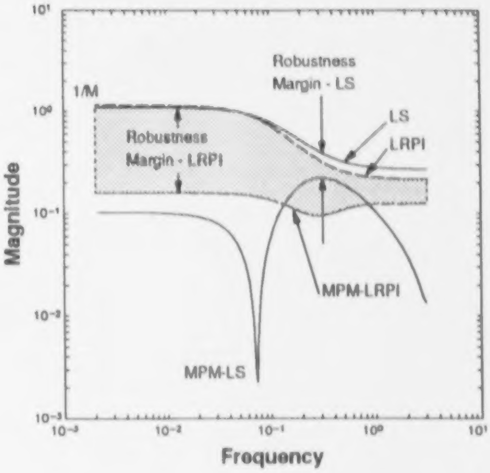


Figure 7.4: A comparison of the robustness margins for the GPC corresponding to the models obtained using the LS and the LRPI methods.

MPM in the mid frequencies. This effect is like an averaging or stretching the MPM for the LS based model.

Note 7.4 *All simulation examples presented in this chapter are based on the following set of tuning parameters.*

$$N_1 = 1,\ N_2 = 10,\ NU = 1,\ \lambda = 0,\ \gamma_\infty = 0,\ P(q) = 1 \ \textit{and}\ C_c(q) = 1 - 0.75q^{-1}.$$

Furthermore, Figure 7.4 shows that the the robustness margin $|1/M(\omega)|$ for the LRPI based model is not significantly different from the LS based model; as a result, the robustness margin for the LRPI case is more or less uniform at all the frequencies as against the LS case where the robustness margin is poor at some frequencies. This example illustrates that the use of LRPI filter significantly enhances the robust stability properties of the GPC controller. The nature of the robustness margin also has a significance effect on the system performance. As an example, the poor robustness margin at certain frequencies for the LS case as shown in Figure 7.4 will lead to a significantly oscillatory reseponse. However the amount of oscillations are significantly reduced for the LRPI case because of uniform robustness margin.

Different servo responses in Figure 7.5 compare the responses between the ideal case,

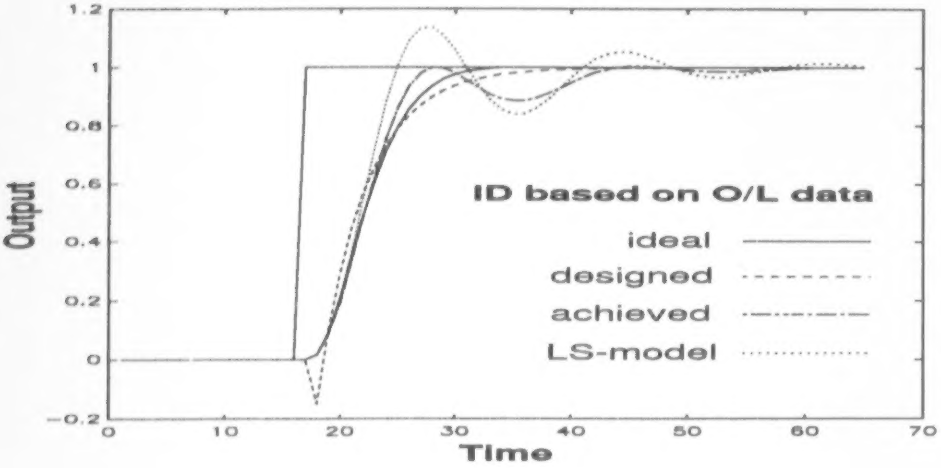


Figure 7.5: Effect of L_s on performance of the system

the designed case, response due to the model obtained using the L_s-filter and the response which is simply based on the model given by equation-(5.52); where:

Note 7.5 *The ideal case corresponds to the ideal situation where the plant dynamics is completely known.*

Note 7.6 *For simplicity reasons, the model corresponding to equation-(5.52) is designated as the LS-model in this chapter.*

Note 7.7 *The designed case corresponds to the block diagram shown in Figure 7.1.*

In Figure 7.5, the designed case corresponds to the model obtained using the L_s-filter. Figure 7.5 illustrates that the response due to the use of L_s-filter does not show any overshoot when compared with the response due to the LS-model. Furthermore, this figure shows that the rise time due to the use of L_s filter is faster than the ideal or the designed cases. However the L_s based response does show some initial oscillatory behavior.

7.4 Control-relevant identification by Rivera *et al.*

A more general single-step strategy to estimate the control-relevant filter was developed by Rivera *et al.* [3, 11], where the objective is to minimize:

$$J^r = \sum_{i=1}^{\infty}[w(i) - y(i)]^2 \tag{7.27}$$

The use of equations-(5.82) to (5.86) in Chapter 5 along with equation-(7.22) yields:

$$J^r = \frac{1 - M(q)\hat{G}(q)}{1 + M(q)\tilde{G}(q)}(R(q)w(t) - S(q)v(t)) \tag{7.28}$$

In the original formulation of the above equation, multiplicative perturbation was used by Rivera *et al.*, whereas here the additive perturbation has been considered. If the system is stable, then from the small gain theorem it is shown that $M(q)\tilde{G}(q) < 1$ (refer equation-(5.44)), which then leads to the following approximation by applying the Taylor series expansion to the denominator in equation-(7.28) as:

$$\frac{1}{1 + M(q)\tilde{G}(q)} \approx 1 - M(q)\tilde{G}(q) + \text{higher-order-terms-neglected} \tag{7.29}$$

Further, using equation-(5.96) it is shown that the designed sensitivity is $\hat{S}(q) = 1 - M(q)\hat{G}(q)$, therefore subsituting this equality along with equation-(7.29) in (7.28) finally gives

$$\begin{aligned} J^r &\approx \hat{S}(q)(1 - M(q)\tilde{G}(q))(R(q)w(t) - S(q)v(t)) \\ &= \hat{S}(q)(R(q)w(t) - S(q)v(t)) - \hat{S}(q)M(q)\tilde{G}(q)(R(q)w(t) - S(q)v(t)) \end{aligned} \tag{7.30}$$

Applying Parseval's theorem and trianglar inequality to equation-(7.30) results in:

$$\begin{aligned} \sqrt{J^r} \;\leq\; & \underbrace{\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{S}(\omega)|^2 |R(\omega)w(\omega) - S(\omega)v(\omega)|^2 d\omega\right]^{1/2}}_{\sqrt{\hat{J}^r}} + \\ & \underbrace{\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{S}(\omega)|^2 |M(\omega)\tilde{G}(\omega)|^2 |R(\omega)w(\omega) - S(\omega)v(\omega)|^2 d\omega\right]^{1/2}}_{\sqrt{J^r_{id}}} \end{aligned} \tag{7.31}$$

The objective in equation-(7.31) is to minimize the function J^r which is bounded at the upper limit by a summation of two terms. The first term $\hat{J}^r$ in equation-(7.31) corresponds to the designed objective which is minimized via the controller design. The effort is therefore made to minimize the second term via suitable model identification, hence this term is designated as the control-relevant identification objective J^r_{id}. As in the case of LRPI, the control-relevant filter $L_r(q)$ is obtained by equating J^r_{id} with J^{OL}_{LS} in equation-(7.25) and by assuming: (a) $\Phi_{uu}(\omega) = 1$; and (b) by neglecting $\Phi_{vv}(\omega)$ in equation-(7.25) to yield:

$$L_r(q) = \hat{H}(q)\hat{S}(q)M(q)(R(q)w(q) - S(q)v(q)) \tag{7.32}$$

The above expression is flexible enough to consider the characteristics of the desired performance (via $\hat{S}(q)$), setpoint and the disturbance to obtain a suitable control-relevant pre-filter

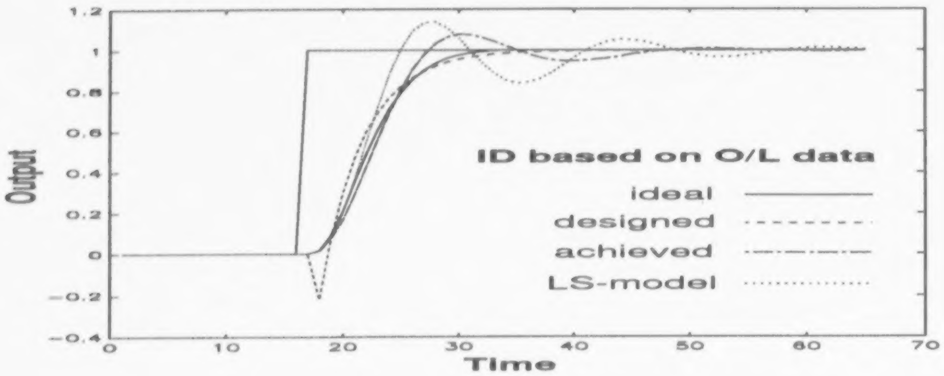


Figure 7.6: Effect of L_r on performance of the system

$L_r(q)$. However for simplicity in the simulation studies (presented in later sections), it is assumed that $v(q) = 0$, no user chosen closed-loop performance specification is provided[11] and it is assumed that step changes in the setpoint are provided *i.e.* $w(q) = 1/\Delta$ in order to estimate $L_r(q)$. When these assumptions are combined with the following relations $\hat{H} = C_c/\hat{A}\Delta$, $\hat{S} = 1 - M\hat{G}$ and $M = S/(\Delta T + \hat{G}S)$ they lead to the following expression for $L_r(q)$:

$$L_r(q) = \frac{C_c(q)\hat{A}(q)T(q)S(q)R(q)}{(\hat{A}\Delta T(q) + \hat{B}(q)S(q))^2}\frac{1}{1-q^{-1}} \tag{7.33}$$

A comparison between the ideal, designed, L_r based achieved and the LS-model based achieved performances are shown in Figure 7.6. This figure shows that the achieved performance due to the use of L_r based filter is in between the ideal/designed cases and the response due to the LS-model. In Figure 7.6 it is also observed that the overshoot for the L_r based response is smaller than the response due to the LS-model. A comparison between the responses due the L_s and L_r filters are shown later in Figure 7.12.

7.5 Control-relevant identification by Zang *et al.*

An iterative control-relevant ID method for the LQG controller was proposed by Zang *et al.* [1, 4, 14] and it is referred as the *Zangscheme*. The method by Zang *et al.* employs closed-loop data of the plant to identify the control-relevant models. In order to facilitate the application of the Zangscheme to GPC, one needs to examine the designed and the achieved closed-loop GPC as shown in Figures 7.7 and 7.8 (note that an equivalent closed-loop GPC was illustrated in Figure 5.2 in Chapter 5). The blocks C_1 and C_2 in Figures 7.7 and 7.8 are given by:

---

[11] The performance specification is naturally provided by the choice of model and the tuning parameters in this case.

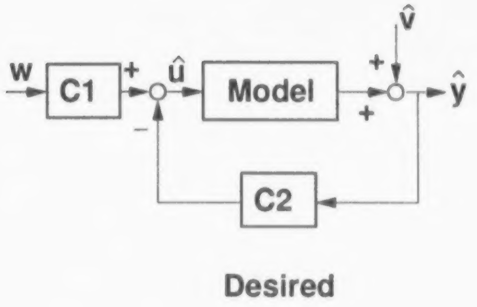


Figure 7.7: Designed closed loop system.

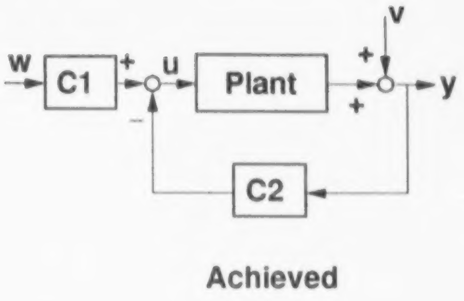


Figure 7.8: Achieved closed loop system.

$$C_1(q) = \frac{R(q)}{\Delta T(q)} \quad \text{and} \quad C_2(q) = \frac{S(q)}{\Delta T(q)} \tag{7.34}$$

The control-relevant ID objective in this case is J_{id} which is given by equation-(7.5) and the control-relevant filter $L_z(q)$ is obtained by equating J_{id} with the closed-loop LS objective J_{LS}^{CL}.

An expression for J_{id} is derived using Figures 7.7 and 7.8, assuming $\hat{v}(t) = 0$. The first step in this sequence is to use equations-(7.20) and (7.22) along with Figures 7.7 and 7.8 to express the achieved and designed input/outputs as (note: in the following and subsequent expressions the argument q is dropped for simplicity):

$$\begin{array}{lll} \text{Designed case:} & \hat{y}(t) = \dfrac{C_1\hat{G}w(t)}{1+\hat{G}C_2} \quad \text{and} & \hat{u}(t) = \dfrac{C_1 w(t)}{1+\hat{G}C_2} \\ \text{Achieved case} & y(t) = \dfrac{GC_1 w(t)}{1+GC_2} + \dfrac{v(t)}{1+GC_2} \quad \text{and} & u(t) = \dfrac{C_1 w(t)}{1+GC_2} - \dfrac{C_2 v(t)}{1+GC_2} \end{array} \tag{7.35}$$

Substituting the expressions in equation-(7.35) in J_{id} in equation-(7.5) and subsequently applying Parseval's theorem gives:

$$\begin{aligned} J_{id} &= \frac{1}{N}\sum_{t=1}^{N}\left[\left\{\frac{GC_1}{1+GC_2}w(t) + \frac{1}{1+GC_2}v(t) - \frac{\hat{G}C_1}{1+\hat{G}C_2}w(t)\right\}^2\right. \\ &\quad \left. + \left\{\frac{C_1}{1+GC_2}w(t) - \frac{C_2}{1+GC_2}v(t) - \frac{C_1}{1+\hat{G}C_2}w(t)\right\}^2\right] \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left\{\frac{|\tilde{G}C_1|^2(1+\lambda|C_2|^2)}{|(1+GC_2)(1+\hat{G}C_2)|^2}\Phi_{ww}(\omega)\right. \end{aligned}$$

$$+ \frac{1+\lambda|C_2|^2}{|1+GC_2|^2}\Phi_{vv}(\omega)\Bigg\}^2 d\omega \tag{7.36}$$

similarly substituting $u(t)$ from equation-(7.35) in the expression for $\epsilon^f(t)$ in equation-(7.23) and taking Parseval's identity leads to:

$$\begin{aligned} J_{LS}^{CL} &= \frac{L}{\hat{H}}(\tilde{G}u(t)+v(t)) \\ &= \frac{L}{\hat{H}}\left(\frac{\tilde{G}C_1 w(t)}{1+C_2G} - \frac{C_2\tilde{G}v(t)}{1+C_2G} + v(t)\right) \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\left|\frac{\tilde{G}(\omega)}{1+G(\omega)C_2(\omega)}\right|^2 |C_1(\omega)|^2\Phi_{ww}(\omega)\right. \\ &\qquad \left. + \left|\frac{1+\hat{G}(\omega)C_2(\omega)}{1+G(\omega)C_2(\omega)}\right|^2 \Phi_{vv}(\omega)\right]\left|\frac{L}{\hat{H}}\right|^2 d\omega \end{aligned} \tag{7.37}$$

Equating the frequency domain expressions for J_{id} in equation-(7.37) with J_{LS}^{CL} in equation-(7.37) and substituting $\hat{G} = \hat{B}/\hat{A}$ and $\hat{H} = C_c/(\Delta\hat{A})$ results in the following relation for the L_z-filter:

$$\begin{aligned} |L_z|^2 &= \frac{|\hat{H}|^2(1+\lambda|C_2|^2)}{|1+C_2\hat{G}|^2} \\ &= \left|\frac{C_cT}{\Delta T\hat{A}+S\hat{B}}\right|^2\left(\frac{|\Delta T|^2+\lambda|S|^2}{|\Delta T|^2}\right) \end{aligned} \tag{7.38}$$

A comparison between the Bode plots of L_s (*i.e.* $\Delta L_s/C_c$), L_r and L_z filters are made in Figure 7.9 which shows that: (a) as expected L_s shows a band-pass characteristics because it is used in conjunction with Δ/C_c; (b) the nature of gain and phase of these filters are significantly different from each other; and (c) for this example problem, L_r and L_z turn out to be a low-pass filter. The achieved servo-response due to the use of L_z-filter is compared with the ideal, designed and the response based on the LS-model in Figure 7.10. This figure shows that the use of L_z filter has helped to reduce the amount of overshoot relative to the LS-model based response.

Different robustness margins due to the use of different filters in Figure 7.11 shows that the nature of MPM associated with these control-relevant models are remarkably different form the MPM that corresponds to the LS-model. Specifically: (a) the L_s filter gives almost a uniform robustness margin but this is achieved by compromising the MPM at the lower frequencies as pointed earlier, (b) the use of both L_r and L_z filters result in smaller MPM at the lower frequencies but it is achieved by compromising MPM at the other frequencies; *e.g.* MPM is high at the mid frequencies due to the use of L_r filter whereas by applying the L_z filter the MPM is significant at the higher frequencies; and (c)

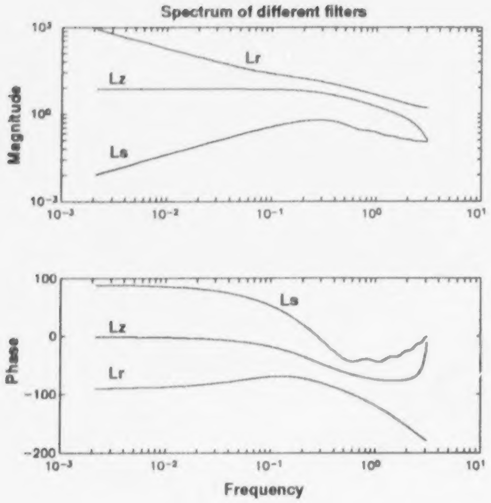


Figure 7.9: Bode plots for the L_s, L_r and the L_z filters.

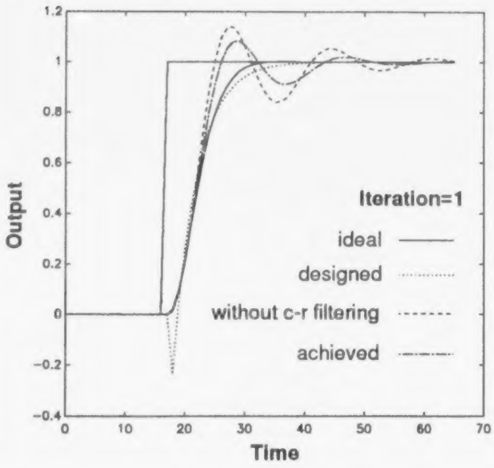


Figure 7.10: Effect of L_z on performance of the system

the robustness bounds are marginally affected bychanges in the identified model due to the use of different filters. The use of L_r and L_z filters give lower MPM at the lower frequencies because they are estimated to be low-pass filters.

Figure 7.11 further shows that the robustness bounds in general are higher at the lower frequencies whereas it goes down with the increase in frequency. Consequently for this case, poor MPM can be accomadated at the lower frequencies relative to the higher frequencies; and interestingly, the use of L_s-filter results in better MPM at the higher frequencies by compromising MPM at the lower frequencies.

The servo responses in Figure 7.12 correspond to the robustness bounds in Figure 7.11 and they show that: (a) the use of an L_r filter results in a remarkably fast rise time but it is achieved at the cost of more overshoot; (b) the use of an L_s filter in this case shows almost not overshoot; and (c) the performance due to the use of L_z filter is somewhere in between the previous two cases. The overshoot in the servo responses due to the L_r and L_z filters can be attributed to the smaller robustness margins at certain frequencies. This example essentially conveys that: ***the uniformity in the robustness margin is probably more important than the accuracy of the model in order to achieve an acceptable performance.***

7.5.1 Weighted objective function and iterative design

An important contribution of the Zangscheme has been to introduce the frequency weightings in the controller design objective which serves the purpose of: (a) accounting for the

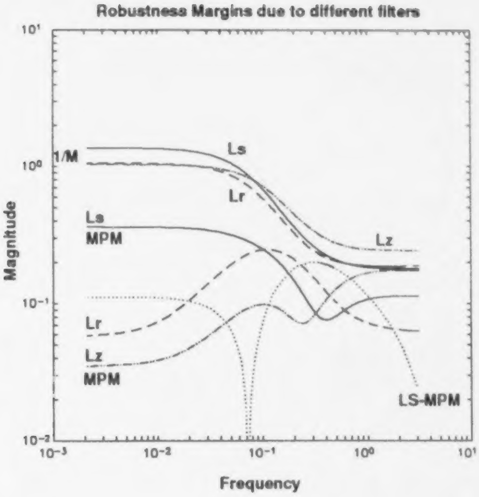


Figure 7.11: A comparison of the robustness margins for GPC due to L_s, L_r and L_z filters.

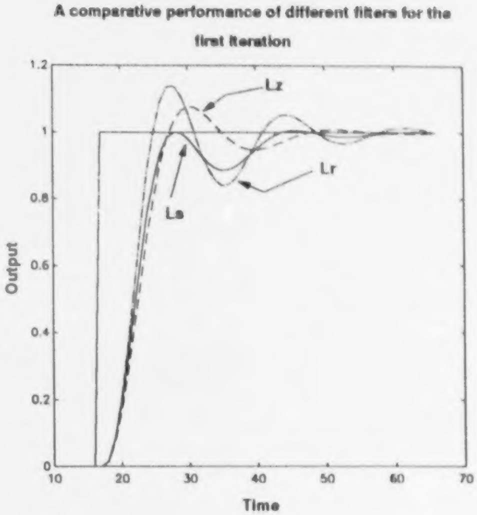


Figure 7.12: A comparison of GPC servo responses due to L_s, L_r and L_z filters.

MPM; and (b) bringing the designed and the achieved controller objectives together. Zang *et al.* [4] proposed the following frequency weighting for the LQG controller design:

$$\hat{J} = \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\{[F_1(z)(\hat{y}(t)-w(t))]^2 + \lambda[F_2(z)\hat{u}(t)]^2\} \tag{7.39}$$

where

$$F_1(\omega) = \sqrt{\frac{\Phi_{(y-w)(y-w)}(\omega)}{\Phi_{(\hat{y}-w)(\hat{y}-w)}(\omega)}} \qquad \text{and} \qquad F_2(\omega) = \sqrt{\frac{\Phi_{uu}(\omega)}{\Phi_{\hat{u}\hat{u}}(\omega)}} \tag{7.40}$$

Zang *et al.* also suggested that the frequency weighting factors $F_1(z)$ and $F_2(z)$ can be adequately modelled by a 3^{rd} order[12] AR models as follows:

$$F_1 = \frac{AR_{(y-w)}(3)}{AR_{(\hat{y}-w)}(3)} \qquad \text{and} \qquad F_2 = \frac{AR_u(3)}{AR_{\hat{u}}(3)} \tag{7.41}$$

The frequency weightings F_1 and F_2 distort the certainty equivalence criterion and Gevers [1] noted the following merits of including these weightings:

---

[12] The choice of model order for F_1 and F_2 is an issue of trade-off. Higher order models will capture the signal spectrum better but it will also lead to an increase in the controller order. Therefore Zang *et al.* suggested a compromize order of 3.

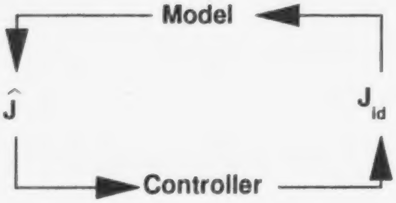


Figure 7.13: Iterative control-relevant design method.

- If at some frequency $\Phi_{(y-w)(y-w)}(\omega) > \Phi_{(\hat{y}-w)(\hat{y}-w)}(\omega)$, (*i.e.* $F_1(\omega) > 1$) it means that at that frequency the model fit is poor, and it contributes to poor tracking performance. Hence in the next design stage more emphasis should be put by the controller to minimize the tracking error due to the mismatch at that frequency.

- If at some frequency $\Phi_{(y-w)(y-w)}(\omega) < \Phi_{(\hat{y}-w)(\hat{y}-w)}(\omega)$ (*i.e.* $F_1(\omega) < 1$), then also the model fit is poor at that frequency, but it helps to achieve a better tracking performance on the actual plant. Therefore the emphasis on the tracking penalty at that frequency is reduced at the next control design stage to provide scope for improvement at other frequencies.

The following steps have been suggested under the Zangscheme to iteratively improve the system performance:

1. For a given (or identified) open-loop model $\hat{G}$, (*i.e.* the LS-model) design a stabilizing controller and perform a closed-loop test on the actual plant as well as for the designed model by subjecting both to the same setpoint (w) profile. The closed-loop data sets for the designed and the achieved case are respectively designated as: $\hat{\mathcal{D}}(N)$ and $\mathcal{D}(N)$.

2. For the present: (a) model $\hat{G}$; (b) controller C; and (c) the data set $\mathcal{D}(N)$, determine the filter L_z. Apply this filter to the present data $\mathcal{D}(N)$ to estimate the next model $\hat{G}$. Design a new controller C based on the new estimated model $\hat{G}$ and by using the present data set $\mathcal{D}(N)$ to determine different frequency weightings.

3. Implement the newly designed controller C on both the plant and model by subjecting both to the same setpoint profile and collect a new set of closed-loop data $\hat{\mathcal{D}}(N)$ and $\mathcal{D}(N)$.

4. Go to step-2 if there is an improvement in the performance or else stop.

The above method is graphically illustrated in Figure 7.13; which essentially conveys the need for updating the model via minimizing J_{id} and use this model to obtain a controller by minimizing $\hat{J}$ iteratively.

The application of this frequency weighting idea has been extended to the GPC in the following sections.

7.6 A GPC algorithm with frequency weightings

A reformulation of the GPC algorithm is proposed in this section by incorporating frequency weighted factors in the GPC objective function. This reformulation in the GPC algorithm is done with a view to enhance its performance using the iterative technique discussed in the previous section.

Let the frequency weighted terms F_1 and F_2 shown by equation-(7.41) be given by:

$$F_1 = \frac{P_n}{P_d} \qquad \text{and} \qquad F_2 = \frac{Q_n}{Q_d} \tag{7.42}$$

and they are used to modify the GPC objective as:

$$J^{\star}_{GPC}(t) = \sum_{j=N_1}^{N_2} \left[\frac{P_n}{P_d}(w(t+j) - Py^{*}(t+j|t))\right]^2 + \lambda \sum_{j=1}^{NU} \left[\frac{Q_n}{Q_d}\Delta u^{*}(t+j-1)\right]^2 \tag{7.43}$$

A series of Diophantine identities can then be defined as follows:

Diophantine identity - 1

$$\frac{P_n}{P_d} = P_{ej} + q^{-j}\frac{P_{fj}}{P_d} \tag{7.44}$$

Diophantine identity - 2

$$\frac{Q_n}{Q_d} = Q_{ej} + q^{-j}\frac{Q_{fj}}{Q_d} \tag{7.45}$$

Diophantine identity - 3

$$\frac{P.P_n}{P_d}\frac{C_c}{\Delta\hat{A}} = E_j + q^{-j}\frac{F_j}{P_d\Delta\hat{A}} \tag{7.46}$$

Diophantine identity - 4

$$\frac{E_j\hat{B}^{\star}}{C_c} = \mathcal{G}_j + q^{-j}\frac{\bar{G}_j}{C_c} \tag{7.47}$$

As a next step in the derivation, equation-(7.44) is multiplied with $w(t+j)$ to yield:

$$w^f(t+j) = P_{e_j} w(t+j) + P_{f_j} w^{ff}(t)$$

where

$$w^f(t+j) = \frac{P_n}{P_d} w(t+j) \quad \text{and} \quad w^{ff}(t) = \frac{w(t)}{P_d} \tag{7.48}$$

Equation-(7.45) is multiplied with $\Delta u^*(t+j-1)$ to give:

$$\Delta u^f(t+j-1) = Q_{e_j} \Delta u(t+j-1) + Q_{f_j} \Delta u^{ff}(t-1)$$

where

$$\Delta u^f(t+j-1) = \frac{Q_n}{Q_d} \Delta u^*(t+j-1) \quad \text{and} \quad \Delta u^{ff}(t-1) = \frac{\Delta u(t-1)}{Q_d} \tag{7.49}$$

Similarly, multiplication of $y^*(t)$ with equation-(7.46) results in:

$$\Psi(t+j|t) = \frac{E_j \hat{B}^\star}{C_c} \Delta u(t+j-1) + F_j y^f(t)$$

where

$$\Psi(t+j|t) = \frac{P P_n}{P_d} y^*(t+j|t) \quad \text{and} \quad y^f(t) = \frac{y(t)}{C_c P_d} \tag{7.50}$$

Substituting equation-(7.47) in equation-(7.50) results in the following predictor:

$$\Psi(t+j|t) = G_j \Delta u^*(t+j-1) + \underbrace{\bar{G} \Delta u^f(t-1) + F_j y^f(t)}_{f^f(t+j)}$$

$$\text{where} \qquad \Delta u^f(t-1) = \frac{\Delta u(t-1)}{C_c} \tag{7.51}$$

The above equation-(7.51) can be compactly written as:

$$\boldsymbol{\Psi} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f}^f \tag{7.52}$$

where $\boldsymbol{\Psi}$ is the predictor vector, $\mathbf{G}$ is the step response matrix (as a result of frequency weighting F_1) and $\mathbf{f}^f$ is the free-response vector.

Substituting equations-(7.48), (7.49) and (7.51) in the objective function given by equation-(7.43) and minimizing it with respect to $\Delta\mathbf{u}$ gives the following after a lengthy simplification process:

$$\Delta \mathbf{u} = \mathbf{H}^{-1} [\mathbf{G}^T \Gamma_y (\underbrace{\mathbf{P}_e \mathbf{w} + \mathbf{P}_f \mathbf{w}^{ff}}_{w^f(t)} - \mathbf{f}^f) - \mathbf{Q}_e \Lambda \mathbf{Q}_f \Delta u^{ff}(t-1)] \tag{7.53}$$

where

$$\mathbf{H} = \mathbf{G}^T \Gamma_y \mathbf{G} + \mathbf{Q}_e^T \Lambda \mathbf{Q}_e^T \tag{7.54}$$

Under the QP framework; the Hessian is **H** and the terms inside the braces [ ] in equation-(7.53) constitute the gradient. The modified GPC law given by equation-(7.53) can also be expressed in the linear feedback form as:

$$T\Delta u(t) = Rw(t) - Sy(t) \tag{7.55}$$

and comparing equation-(7.55) with (7.53) gives:

$$\begin{aligned} T &= C_c \mathbf{P}_d \mathbf{Q}_d + \mathbf{P}_d \mathbf{Q}_d \mathbf{h} \mathbf{G}^T \Gamma_y \bar{\mathbf{G}} + C_c \mathbf{P}_d \lambda \mathbf{h} \mathbf{Q}_e \mathbf{Q}_f \\ R &= C_c \mathbf{Q}_d \mathbf{P}_n \mathbf{h} \mathbf{G}^T \Gamma_y \\ S &= \mathbf{Q}_d \mathbf{h} \mathbf{G}^T \Gamma_y \mathbf{F} \end{aligned} \tag{7.56}$$

where **h** in equation-(7.56) is the first row of $\mathbf{H}^{-1}$. Kwok and Shah [18] extended the GPC law to include steady-state weighting to enhance its stability property. This result can also be applied to the modified GPC law as shown below:

$$\begin{aligned} e_s &= \frac{\mathbf{P}_n(1) C_c(1)}{\mathbf{P}_d(1) \hat{A}(1)} \\ g_s &= \frac{\hat{B}^\star(1) \mathbf{P}_n(1)}{\hat{A}(1) \mathbf{P}_d(1)} \\ F_s &= e_s \mathbf{P}_d \hat{A} \\ \bar{G}_s &= g_s \frac{T}{\Delta} - e_s \frac{\hat{B}^\star}{\Delta} \end{aligned} \tag{7.57}$$

where e_s, g_s, F_s, and $\bar{G}_s$ are the values of E_j, G_j, F_j, $\bar{G}_j$ at the steady state respectively.

The application of modified GPC to iteratively enhance its performance is described in the following sections

7.6.1 Iterative implementation of L_s

The use of L_s filter in conjunction with the modified GPC to iteratively upgrade the process performance is discussed in this subsection. The loss function plots in Figure 7.14 shows that under the influence of weighting functions F_1 and F_2, the performance of modified GPC improves. The loss functions in this figure corresponds to the expressions given by equation-(7.9). It is also seen in this figure that: (a) in the beginning the designed performance is better than the achieved performance as expected; but (b) with the progress in iteration-steps, the achieved performance becomes better than the designed performance

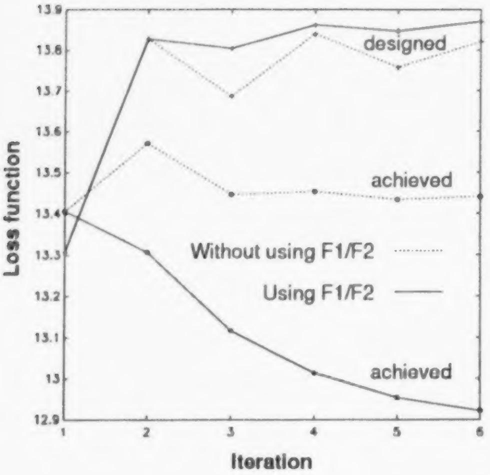


Figure 7.14: The influence of L_s on the progress of designed-vs-achieved performance criteria along the iteration steps.

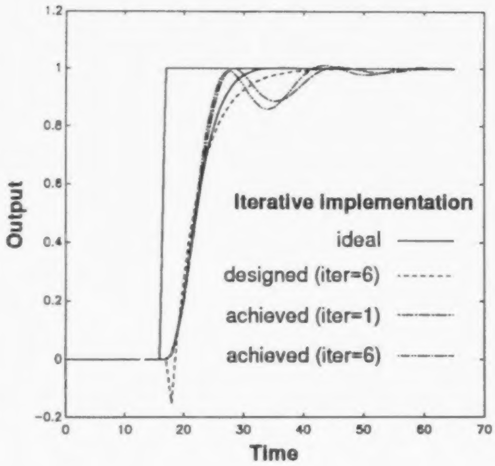


Figure 7.15: The effect of L_s on the achieved performance at iterations 1 and 6.

and also the overall achieved performance improves along the line. However this figure shows that the improvement in the achieved performance in terms of the relative decrease in the loss function is only by 3.8% between the iteration steps 1 and 6. This observation is also reflected in the corresponding servo responses in Figure 7.15; which shows a marginal improvement in the achieved performance with the change in iteration steps from 1 to 6. Moreover, this improvement in the servo response is due to a marginally faster rise time corresponding to iteration number 6. The advantage of including the weighting functions F_1 and F_2 on the servo-response is depicted in Figure 7.16, which shows a better response is obtained when these weighting functions are considered.

The evolution of Bode plots for the identified models using the L_s-filter are illustrated in Figure 7.17; which shows that there is no significant change in the model with progress in the iteration steps. This figure further shows that the magnitude spectrum of the model fits poorly in the high frequency range compared to the LS-model, whereas the phase spectrum of the model fits better relative to the LS-model at the high frequencies. A more accurate description of the model fit can be given via the MPM plot, which is subsequently shown in Figures 7.19 and 7.20.

The Bode plot in Figure 7.18 corresponds to the evolution of $\Delta L_s/C_c$ with the progress in iterations. This figure shows that there is virtually no change in the nature of $\Delta L_s/C_c$ along the iteration steps; this happens because the change in the model is marginal

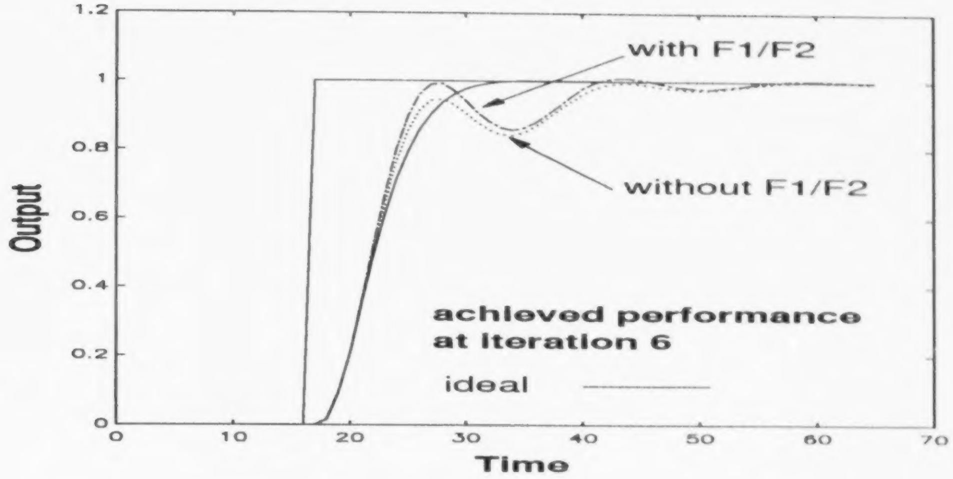


Figure 7.16: The effect of frequency weightings F_1 and F_2 on the achieved performance with the use of L_s-filter at iteration-6.

as apparent in Figure 7.17; and the tuning parameters are kept constant.

The evolution of robustness margin with the progress in iterations are shown in Figures 7.19 and 7.20. Also shown in these figures are the spectrum of the weighting functions F_1 and F_2. These figure shows that with the use of L_s filter, the uniformity in the robustness margin is maintained more or less along the iteration steps. However the MPM plots in these figures indicate that the mismatch at the lower frequencies increases with the progress in iterations, which however is not a matter of great concern because the robustness bound is sufficiently large at these frequencies.

Earlier in subsection 7.5.1 it was remarked that if $F_1(\omega) > 1$ at any frequency, then the MPM at that frequency is responsible for poor tracking performance [1]. In connection with this, Figure 7.19 shows that the magnitude spectrums of F_1 and F_2 are significantly higher than 1 at a certain frequency range that is marked by a shaded box. Therefore according to the earlier remark, the MPM in this shaded region is mostly responsible for the degradation in the system performance. Furthermore, it is interesting to note that the robustness margin is narrow in this shaded frequency range relative to other frequencies; which is chiefly responsible for deteriorating the system performance, according to the discussions in Chapters 5 and 6. Thus we see that the explanations provided for the reasons behind deteriorating sytem performance can be viewed form two different angles; and they appear to be in agreement.

The characteristics of F_1 and F_2 at iteration 1 is utilized by the modified GPC

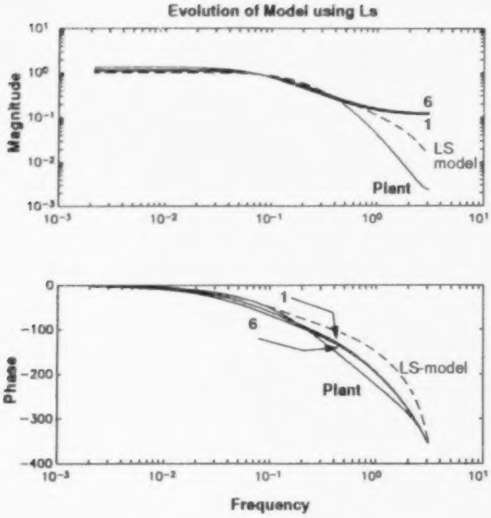


Figure 7.17: Evolution of the estimated models by using L_s.

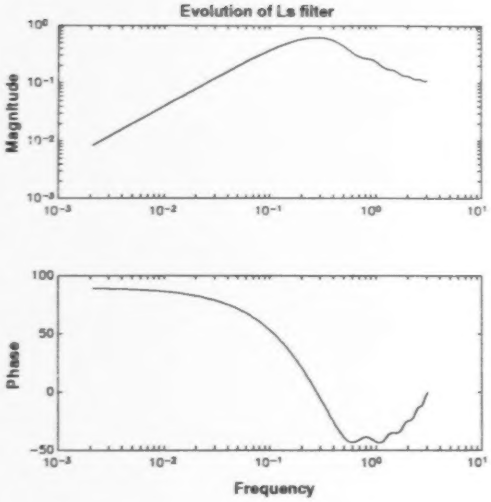


Figure 7.18: Evolution of $\Delta L_s/C_c$ with the progress in iteration.

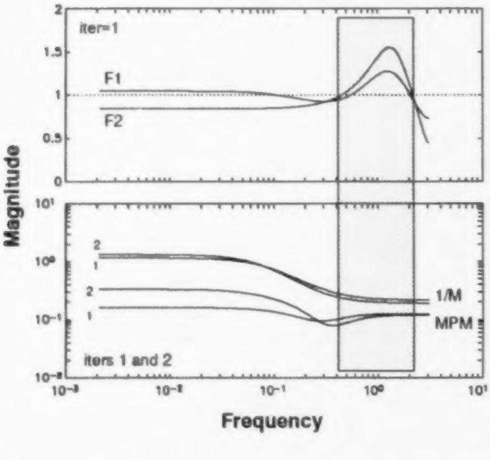


Figure 7.19: The frequency weighting functions F_1 and F_2 at iteration 1 and the robustness margins at iterations 1 and 2 with the use of L_s-filter.

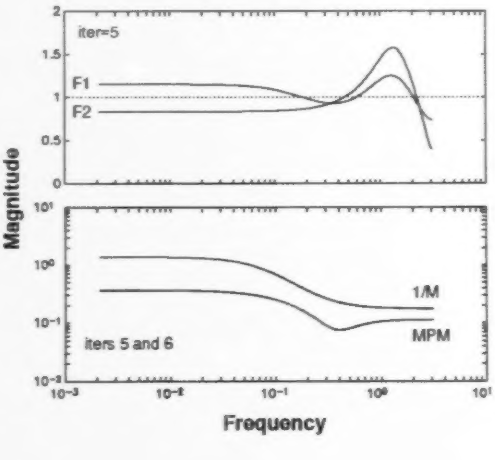


Figure 7.20: The frequency weighting functions F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6 with the use of L_s-filter.

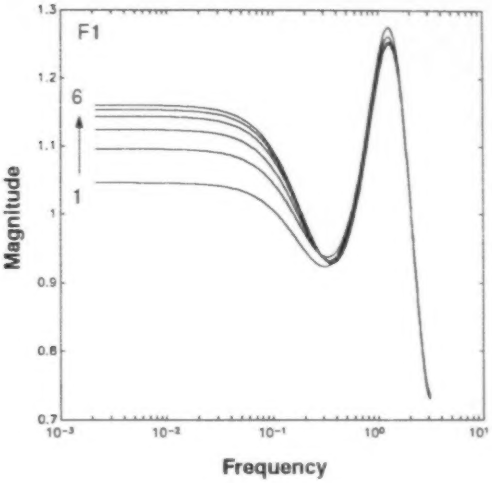


Figure 7.21: The evolution F_1 for the case of L_s-filter.

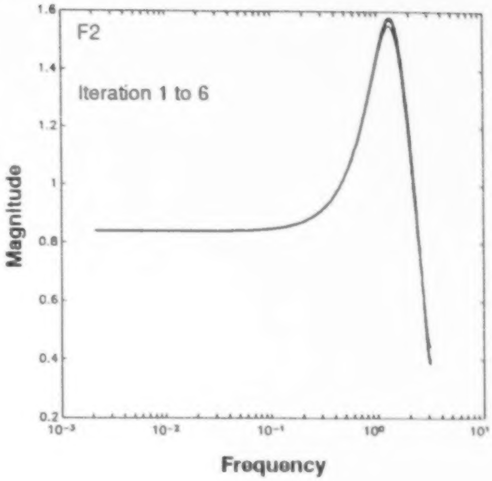


Figure 7.22: The evolution of F_2 for the case of L_s-filter.

to improve its closed-loop performance in the next iteration. For example, the robustness margin obtained for the second iteration in Figure 7.19 shows that the MPM is reduced at the shaded frequency region by compromising at the lower frequencies.

Figure 7.20 shows the magnitude spectrum of F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6. It is seen in Figure 7.20 that there is no apparant change in the robustness margins between iterations 5 and 6, which means that, for the specified model structure, no further improvement in the system performance is possible, which is also conveyed by the loss function plots in Figure 7.14. Figure 7.20 further shows that the peaks in the spectral plots of F_1 and F_2 are still existing at iteration 5 around the same frequencies as marked by the shaded area in Figure 7.19; which implies that for the specified model structure and probably for the specified L_s-filter, no further improvement in the system performance is possible and the performance problems for this example case will manifest mostly from the MPM in the shaded frequency range. A similar explanation can also be provided by using the robustness margin plots.

The evolution of F_1 and F_2 weighting functions along the iteration steps are similarly portrayed in Figures 7.21 and 7.22 respectively. Figure 7.22 shows that there is virtually no change in the pattern of F_2, with the progress in the iteration steps, which indicate that F_2 does not contribute in the enhancement of the achieved performance in conjunction with the L_s-filter. However with the use of L_s-filter, the characteristics of the F_1 weighting changes with the iteration steps and it indicates that: (a) its value increases in the lower frequency range with the increase in iteration steps; but (b) it does not effect the nature of peak in a major way.

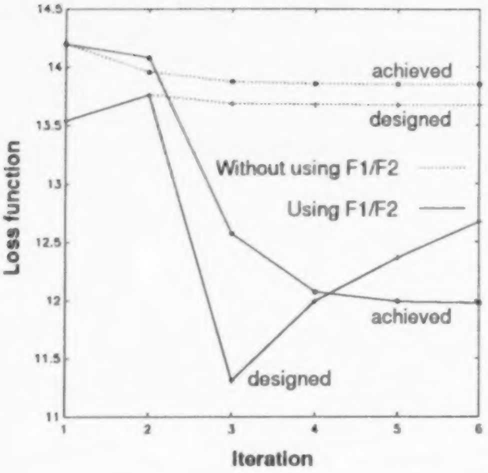


Figure 7.23: The influence of L_r on the progress of designed-vs-achieved performance criteria along the iteration steps.

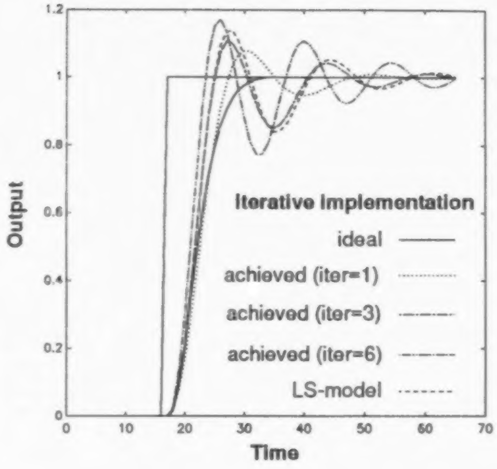


Figure 7.24: The effect of L_r on the achieved performance at iterations 1, 3 and 6.

7.6.2 Iterative implementation of L_r

This section discusses the influence of L_r-filter on the performance of modified GPC when implemented iteratively. Figure 7.23 shows the effect of frequency weightings F_1 and F_2 on the designed and achieved closed-loop performances; which illustrates that the use of frequency weightings significantly improves the performance of GPC when iteratively. For the frequency weighted case, this figure shows that, the loss-function values for both the designed and the achieved performances drop till a certain iteration step and thereafter the achieved performance improves relative to the designed performance. This observation is somewhat reflected in the servo responses in Figure 7.24, which shows an acceleration in the rise time with the progress in iteration steps and this way it gives lower loss function values. Faster rise time is however achieved at the cost of more overshoot, for example, at the 6^{th} iteration, the rise time is fastest but it manifests as an overshoot of around 19%. Therefore just from the loss-function plots, the quality of the response cannot be ascertained. Nevertheless, for this referred example together with the use of L_r-filter, it is seen that the loss-function of the achieved-loop drops by about 18.3% in six iteration steps; which is a significant improvement.

The effect of including the frequency weightings F_1 and F_2 are depicted in Figure 7.25, which shows that these weightings significantly influence the system performance. For the unweighted case, this figure shows that the overshoot is smaller relative to the weighted

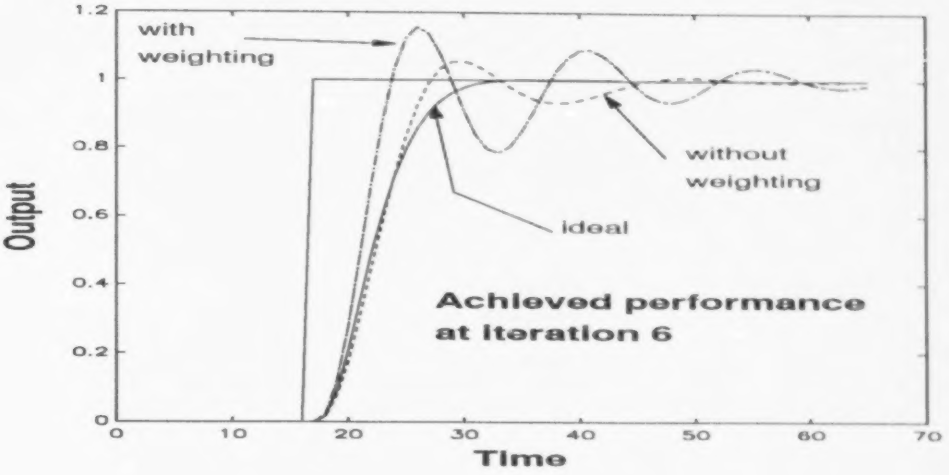


Figure 7.25: The effect of frequency weightings on the achieved performance with the use of L_r-filter at iteration 6.

case.

Figures 7.26 and 7.27 respectively illustrate the evolution of model and the L_r-filter along the iteration steps. The magnitude plots in Figure 7.26 show that the mismatch is significant at the higher frequency range relative to the LS-model. However with the progress in iteration steps, attempt is made to reduce this gain mismatch at the higher frequencies. In case of the phase spectrum, it is noticed that the L_r based models are closer to the plant when compared with the LS-model. However unlike the gain, the phase spectrum tends to move away from the plant with the progress in iterations.

Figure 7.27 shows that the L_r filter evolves in a significant manner along the iteration steps. At the first iteration, in Figure 7.27 it is noticed that L_r has a strong low-pass characteristics, as a result it identifies the plant better around the steady state by compromising at the higher frequencies. However along the iteration steps, this low-pass characteristics changes which has a significant impact on the identified model, which are further examined in Figures 7.28 to 7.31 via the robustness margin plots.

Figures 7.28 to 7.31 depict the evolution of robustness margins and the frequency weighting factors. Figure 7.28 shows that better model estimate is obtained at the lower frequencies by significantly compromising the model fit at the higher frequencies for the iteration step 1. This happens because of the strong low-pass nature of the L_r-filter in the initial stage. This figure also shows that the magnitude spectrums of F_1 and F_2, exhibit a significant peak around the same frequencies as in case of the L_s-filter. Corresponding to

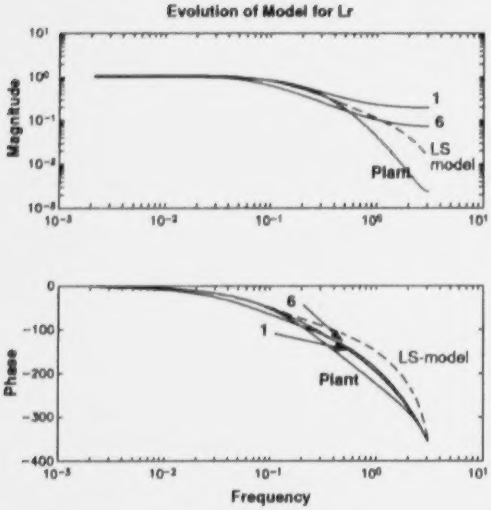


Figure 7.26: Evolution in the estimated models by using L_r.

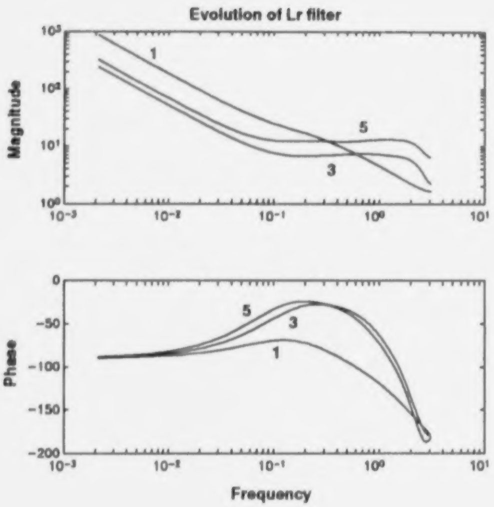


Figure 7.27: Evolution of L_r with the progress in iteration.

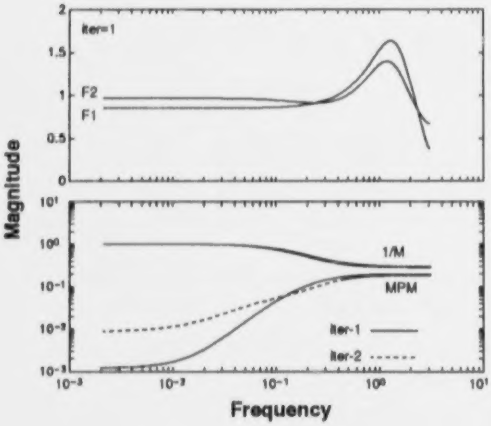


Figure 7.28: The frequency weighting filters F_1 and F_2 at iteration 1 and the robustness margins at iterations 1 and 2 for the case of L_r-filter.

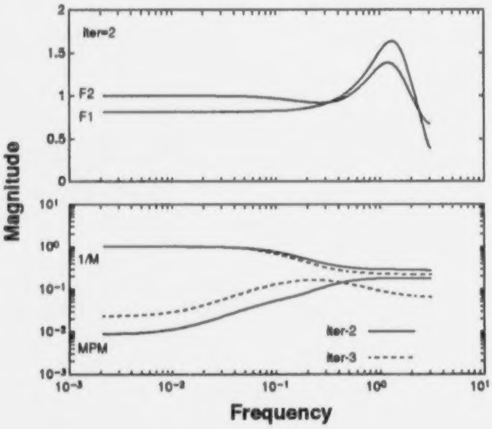


Figure 7.29: The frequency weighting filters F_1 and F_2 at iteration 2 and the robustness margins at iterations 2 and 3 for the case of L_r-filter.

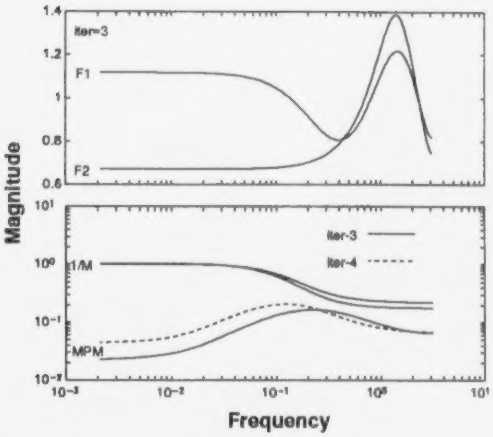


Figure 7.30: The frequency weighting filters F_1 and F_2 at iteration 3 and the robustness margins at iterations 3 and 4 for the case of L_r-filter.

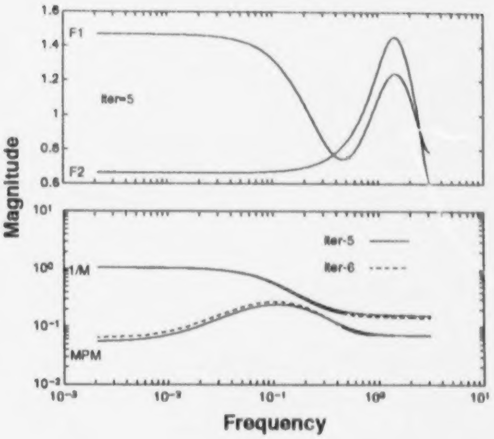


Figure 7.31: The frequency weighting filters F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6 for the case of L_r-filter.

this 'peak' frequency range, it is observed that the robustness margin is also poor. Therefore in the second iteration, effort is made by the control-relevant identifier to obtain a better estimate of the plant around the peak frequency range by compromising the model fit in the lower frequency range as depicted in this figure for the iteration step 2.

In Figure 7.29 it is seen that the nature of F_1 and F_2 is similar to the previous case, but the MPM corresponding to the third iteration shows some improvement around the peak frequency range, but the narrowness in the robustness margin now shits to an adjacent frequency range.

At the third iteration, Figure 7.30 shows that the characteristics of F_1 changes significantly at the lower frequencies; but this also leads to an improved robustness margin around the peak frequencies for the subsequent fourth iterative step. This improvement in the robustness margin is achieved by shifting the MPM peak to an adjacent frequency. At iteration 5, it is seen in Figure 7.31 that the frequency weightings does not change the robustness margin for the subsequent stage 6 in a significant manner. However during the fifth or sixth iteration, the robustness margin becomes narrow at the middle frequencies, thus indicating that the system can become unstable with further progress in the iteration steps.

The evolution of magnitude spectrum for the weighting functions F_1 and F_2 are

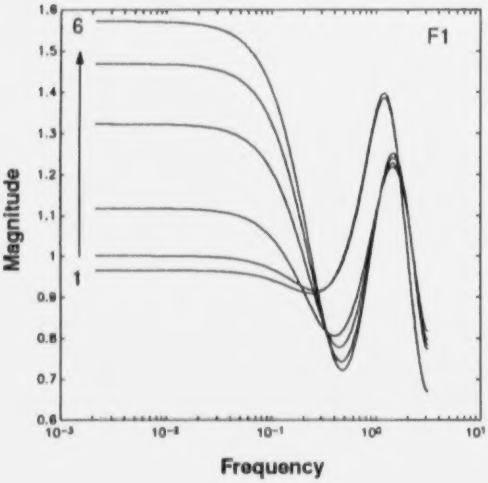


Figure 7.32: The evolution F_1 for the case of L_r-filter.

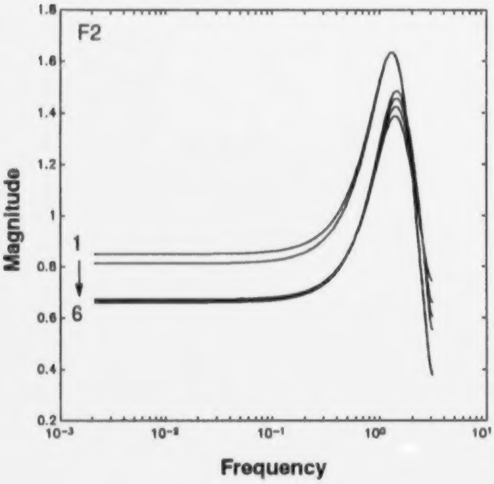


Figure 7.33: The evolution of F_2 for the case of L_r-filter.

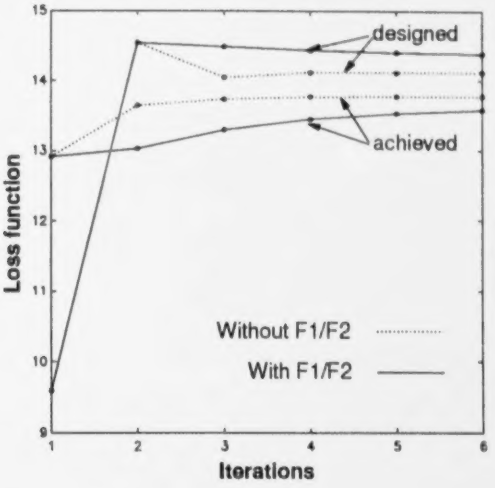


Figure 7.34: The influence of L_z on the progress of designed-vs-achieved performance criteria along the iteration steps.

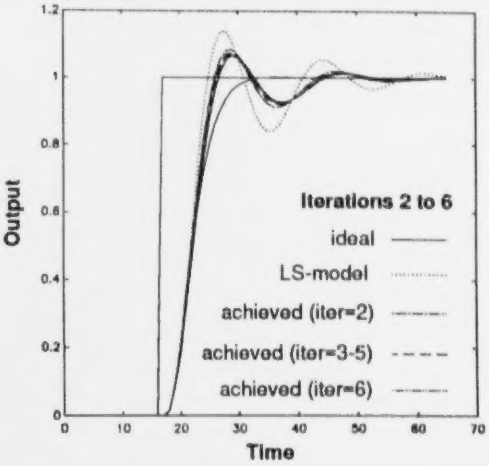


Figure 7.35: Achieved performances at different iterations for the case of L_z-filter.

shown respectively in Figures 7.32 and 7.33. This figure shows that, unlike the case of L_s filter, here both the weighting functions change along the iteration steps; and they both influence the controller design.

7.6.3 Iterative implementation of L_z

The effect of L_z-filter in the performance of modified GPC when implemented iteratively is discussed in this section. Figure 7.34 shows the progress in the designed and achieved performance objectives (or loss function) along the iteration steps. This figure also shows the consequence of including and not including the the frequency weightings F_1 and F_2 for the iterative implementation. In Figure 7.34, the iteration step 1 corresponds to the closed-loop simulation which is based on the LS-model. The loss function plots in this figure show that in the beginning as expected, the designed performance is better than that achieved. But in the subsequent iterations the achieved objective performs better than the designed objective.

Nevertheless, with the progress in iteration the achieved performance does not improve and on the contrary this value increases. Gevers noted that such a situation may be possible and this issue is currently being investigated by him [1]. The other reason for the non-improvement in the achieved objective could be because the derivation of the L_z-filter which is for the LQG objective is not suitable for the GPC objective. The responses in Fig-

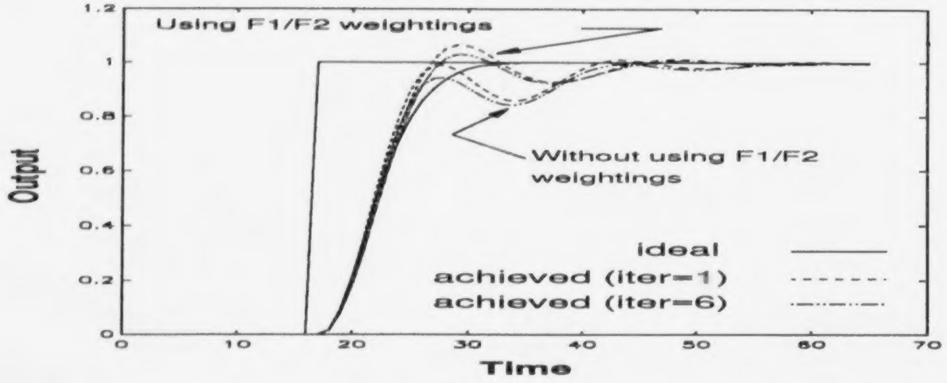


Figure 7.36: The effect of frequency weightings on the achieved performance with the use of L_z-filter at iteration-6.

ure 7.36 show that the weighting functions favorably influence the closed-loop performance.

Figure 7.37 describes the evolution of estimated model using the L_z-filter. This figure shows, as in the prevous cases of filters, the mismatch in gain is significant at the higher frequencies, whereas the phase is closer to the plant relative to the LS-model. The evolution of the L_z-filter along the iteration steps is depicted in the form of Bode plots in Figure 7.38; which shows that L_z has low-pass characteristics.

The evolution of robustness margins for different iterations are shown in Figures 7.39 and 7.40. In Figure 7.39 it is seen that profile of the L_z based estimated MPM changes significantly compared to the LS-model. This figure also shows that for the L_z based model, the robustness margin is small in the frequency range where the peak in F_1 and F_2 are significant. Figure 7.40 shows no apparant change in the robustness margin between iterations 5 and 6, signifying that the loss function has reached a plateau. The evolution of F_1 and F_2 along the iteration steps are shown in Figures 7.41 and 7.42 respectively.

The evolution of the robustness margin with the progression in the iterative steps are shown in Figures 7.39 and 7.40. This figure shows a significant change in the MPM spectrum when compared with the initial design stage. The MPM corresponding to the 3^{rd} or 5^{th} iteration shows a more uniform distribution, unlike the MPM for the open-loop model. However it is also seen for this example that, at the higher frequencies the robustness margin reduces with the progress in iteration and it is expected that with the further progress in iteration, this robustness margin may get violated and result in instability.

The evolution of model with the successive iterations is shown in Figure 7.37. This figure shows that the mismatch in the magnitude at the higher frequencies increases with the increase in iteration. However the phase tries to catch-up with the plant along the iteration steps. Figure 7.38 shows how the Bode plot of the L_z filter evolves along the

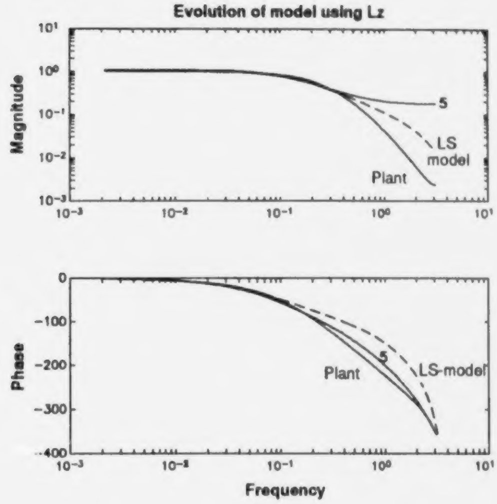


Figure 7.37: The effect of L_z on the evolution of model.

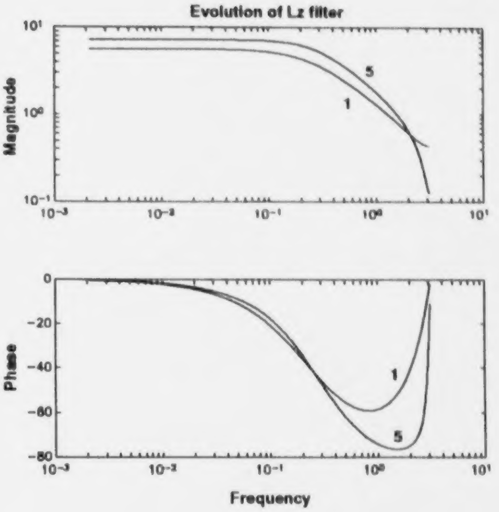


Figure 7.38: The evolution of L_z along the iteration steps.

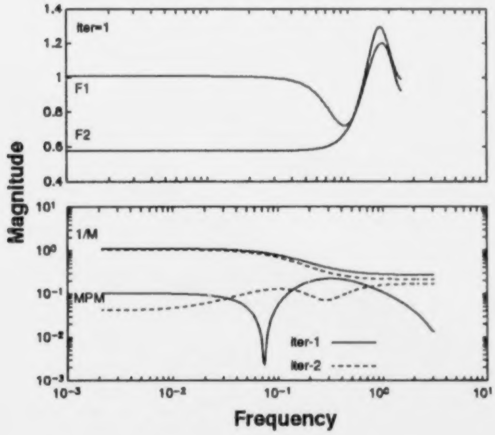


Figure 7.39: The frequency weighting filters F_1 and F_2 at iteration 1 and the robustness margins at iterations 1 and 2 for the case of L_z-filter.

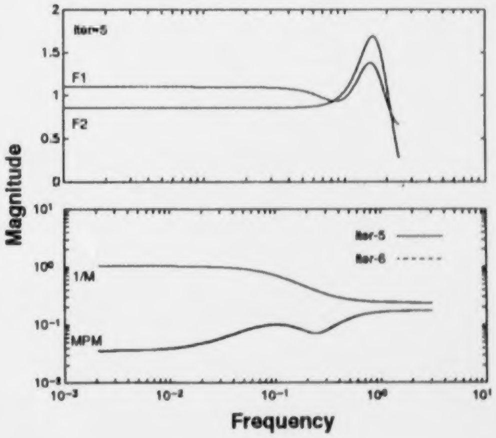


Figure 7.40: The frequency weighting filters F_1 and F_2 at iteration 5 and the robustness margins at iterations 5 and 6 for the case of L_z-filter.

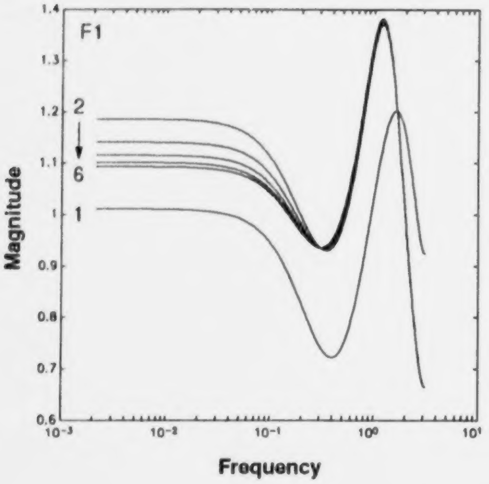


Figure 7.41: The evolution F_1 for the case of L_z-filter.

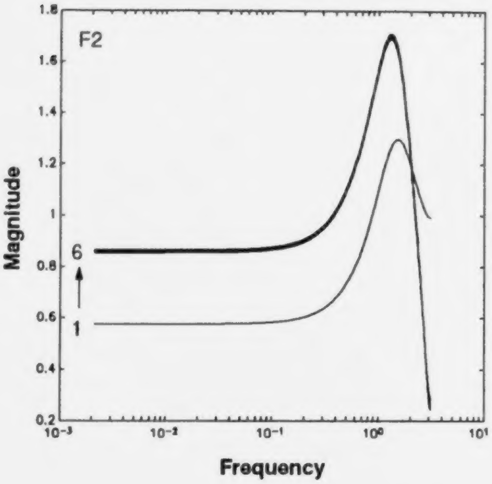


Figure 7.42: The evolution of F_2 for the case of L_z-filter.

iteration steps.

The comparative achieved performance due to using different filters is displayed in Figure 7.43 and their corresponding servo responses at the last iteration is depicted in Figure 7.44. It is seen that according to the loss function plots in Figure 7.43, the use of L_r-filter gives the best performance in terms of achieving minimum values for the loss functionbut its corresponding response in Figure 7.44 shows larger overshoot relative to the other two cases.

In case of the L_s-filter, the reduction in the loss function is gradual but it does not give any overshoot. This figure also shows that the performance due to the use of L_z filter is close to the response based on the LS-model.

7.7 Conclusions

Control relevant identification methods due to Shook and Shah, Rivera *et al.* and Zang *et al.* have been reviewed and were applied to GPC. These control-relevant methods are an approximation of the dual control, hence they are suboptimal in nature. The objective of these control relevant ID methods is to improve the closed-loop behavior by formulating an ID objective that is commensurate with the controller goal. The key step in these control-relevant ID methods is the formulation of the control-relevant filters, which is obtained by equating the identification and the LS objectives. Control-relevant models are obtained by

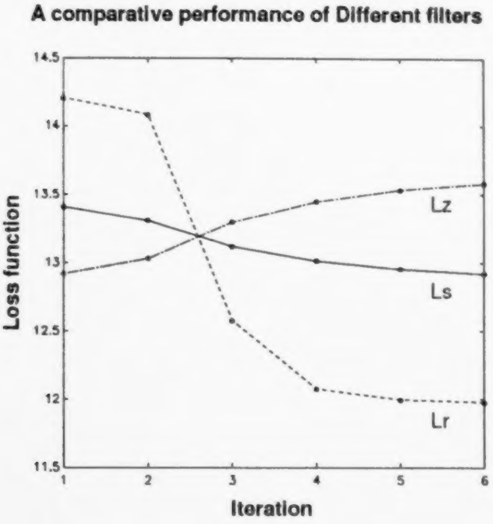


Figure 7.43: A comparison of different achieved performances obtained using different control-relevant filters.

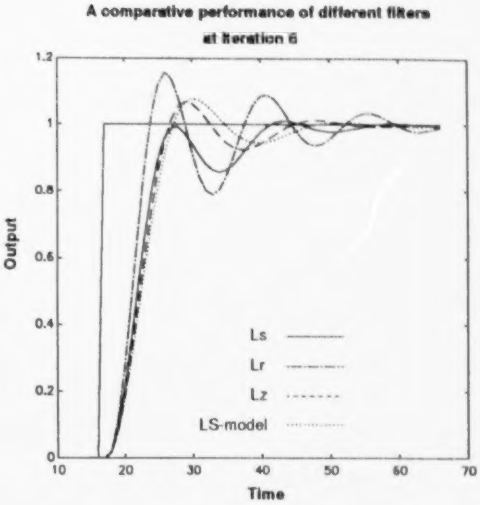


Figure 7.44: A comparison of servo responses obtained using different control-relevant filters.

first filtering the process data with the control-relevant filters and then applying the LS technique. The idea of frequency weighting due to Zang *et al.* have been used to modify the GPC control law. These control-relevant filters have been tested by iteratively assessing the performance of the modified GPC. These simulation results indicate the following:

- The use of an L_r-filter provided the minimum loss function values for the achieved performance. This was accomplished by having a faster rise time but it also resulted in a more overshoot. The robustness margin for this case was not uniformly distributed.
- The use of L_s filter also helped to improve the performance but its iterative implementation improved the performance only marginally. The use of an L_s filter in general provided a more uniform robustness margin relative to the other two cases; and consequently it did not show any significant overshoot.
- The L_z-filter did not appear to work satisfactorily for the modified GPC.

Bibliography

[1] M. Gevers. *Towards a Joint design of Identification and Control?* In H.L.Trentelman and J.C.Willems (Eds), Essays on Control: Perspectives in the Theory and its Applications, Birkhäuser, Boston., 1993.

[2] D.S. Shook, C. Mohtadi, and S.L. Shah. "A Control-Relevant Identification Strategy for GPC". *IEEE Trans. on AC*, 37(7):975 - 980, 1992.

[3] D.E. Rivera, J.F. Pollard, and C.E. Garcia. "Control-Relevant Prefiltering: A Systematic Design Approach and Case Study". *IEEE Trans. on AC*, 37(7):964 - 974, 1992.

[4] Z. Zang, R.R. Bitmead, and M. Gevers. "H_2 Iterative Model Refinement and Control Robustness Enhancement". In *Proceedings of the* 30^{th} *CDC Meeting, December 1991*, Brighton, England, 1991.

[5] B.D.O. Anderson and R.L. Kosut. "Adaptive Robust Control: on-line learning ". In *Proceedings of* 30^{th} *IEEE Conf. on Decision and Control*, pages 297–298, Brighton, UK., 1991.

[6] A.A. Fel'dbaum. "The theory of Dual Control, Pts: 1-4". *Automation and Remote Control*, 1960-61. This reference is from [1].

[7] A.A. Fel'dbaum. *Optimal Control Theory.* Academic Press, New York, USA, 1965.

[8] D.S. Shook. *Implementation Issues in Adaptive GPC.* Ph.D. Thesis, Department of Chemical Engineering, University of Alberta, Edmonton, Canada., 1992.

[9] K.J. Åström and B. Wittenmark. "Problems of Identification and Control". *Journal of Mathematical Analysis and Applications*, 34:90 - 113, 1971.

[10] K.J. Åström and B. Wittenmark. *Adaptive Control.* Addison-Wesley, Reading, Mass., USA, 1989.

[11] D.E. Rivera, J.F. Pollard, L.E. Sterman, and C.E. Garcia. "An Industrial Perspective on Control Relevant Identification". In *American Control Conference*, pages 2406–2411, San Diego, California, USA, 1990.

[12] D.E. Rivera, K.S. Jun, E. Elisante, and V.E. Sater. "Robustness and identification issues in horizon predictive control with application to a binary distillation column". In *American Control Conference*, pages 3031–3035, Chicago, USA, 1992.

[13] R.J.P. Schrama. "Accurate Identification for Control: The Necessity of an Iterative Scheme". *IEEE Trans. on AC*, 37(7):991 – 994, 1992.

[14] Z. Zang, R.R. Bitmead, and M. Gevers. "Iterative Weighted Least-squares Identification and Weighted LQG Control Design". *Automatica*, 31(11):1577 – 1594, 1995.

[15] J.T. Oden. *Applied Functional Analysis*. Prentice Hall, New Jersey, USA, 1979.

[16] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. "Generalized Predictive Control-Part I and II". *Automatica*, 23:137–160, 1987.

[17] L. Ljung. *System Identification-Theory For The User*. Prentice Hall, New Jersey, USA, 1987.

[18] K.Y. Kwok and S.L. Shah. "Long-Range Predictive Control with a Terminal Matching Condition". *Chem. Eng. Sci.*, 49(9):1287 – 1300, 1994.

Chapter 8

Conclusions, recomendations and research directions

8.1 Concluding remarks

Robust tuning guidelines for GPC and Markov-Laguerre based MPC have been proposed by integrating results from signal processing, system identification, robust control and LRPC design. The small gain theorem was used as a tool to carry out the robust design of LRPC. Robust design methods require a spectrum of the model-plant mismatch (MPM) that is associated with the identified model, but it is not concerned how that MPM is estimated. Signal processing methods were used to fill this gap by estimating the spectrum of the MPM from the process data. Another issue related to robust design is performance, which was also addressed in this thesis. In the context of performance enhancement, different control relevant identification methods have been examined and evaluated via applications using to GPC. The key issues in performance enhancement are the design of appropriate control relevant filters and modifications in the control objective functions; and using this information to upgrade the process model.

8.1.1 Main contributions

The main contributions of this thesis are: (a) a methodology for the robust design of LRPC; and (b) design methods for performance improvement (of GPC). Besides, there are also a number of other results which in themselves are interesting contributions. The important contributions of this thesis are summarized below:

1. Signal processing

A number of time and frequency domain based signal processing results that are relevant to system identification have been reviewed with a tutorial flavor and their application highlighted via example dependent case studies and illustrations. In the area of time domain signal processing, the use of different correlation functions with confidence limits has been demonstrated by estimating the process and noise characteristics. It is shown how the time domain process data can be used to effectively estimate the spectral characteristics of the signal and the process (*i.e.* the Bode plots) with a calculated level of confidence, by using the DFT (*i.e.* FFT) in conjunction with signal processing based smoothing methods. For noisy process data, it has been shown how the coherency spectrum can be used as a reliability index in estimating the Bode plots. These spectral techniques were also used to estimate the spectrum of the MPM from open loop plant data.

2. System identification

Different components of system identification (ID) *i.e.*: (a) experiment design; (b) model estimation methods; and (c) model validation have been reviewed with a tutorial flavor and in the context of estimating the transfer function (*i.e.* ARX, BJ) and Markov-Laguerre models from process data. The estimation techniques considered were: LS, AUDI and PEM methods. For model validation, the time domain signal processing methods were evaluated and recommended. The use of these ID methods was successfully illustrated with the: (a) the Shell bench mark problem; and (b) an industrial process. The key features in the Shell problem were: (i) the process consisted of a 2×2 triangular system; (ii) response in two of the channels was instantaneous whereas there was a delay of 6 units in the remaining channel; and (iii) the process was heavily corrupted by AR type noise. System ID methods were successfully used to ascertain these features. For the second industrial process, estimation of the delay was a key issue. This delay was estimated by fitting the model in the frequency domain.

Orthonormal function models were introduced with a historical perspective. Certain mathematical premises were used to define the scope of orthonormal function models. The AUDI method was used to simultaneously estimate parameters of all 1 to N^{th} order orthonormal function models in one computational step. It was shown how the proposed method complements the LS_y solution by verifying an appropriate order. The proposed method was verified using Laguerre models with real and complex poles, Kautz model, FIR/step response models and a Markov-Laguerre model.

3. Robust design of GPC

The small gain theorem was used to provide robust tuning guidelines for GPC. It was shown that the robustness bound of GPC is influenced most significantly by the model at lower frequencies. At higher frequencies the robustness bound is influenced by the controller term, provided that the model rolls off at these higher frequencies. The influence of different tuning parameters of GPC on the robustness bound were examined via simulations and experimental evaluations. For a first order model, explicit expressions were derived for the GPC Diophantine coefficients, GPC control law and linear GPC polynomials. These expressions were used to analytically ascertain the robustness properties of N_2, λ and $C_c(q)$ tuning parameters. A method was presented to estimate the spectrum of the MPM and to determine the model from closed loop data. This method allows one to examine the robustness of GPC quasi-adaptively. An optimization method was also proposed to select some of the GPC tuning parameters within the framework of small gain theorem.

4. Robust design of Markov-Laguerre based MPC

A Markov-Laguerre model was used to formulate a MPC by first converting this model into a state space form. The objective function of this MPC was the same as GPC. The concept of steady state error weighting was incorporated in this MPC to enhance the stability. A structured noise model was combined with the Markov-Laguerre model to obtain an MPC that resulted in faster disturbance rejection. As in the case of GPC, the SGT was used to evaluate the effect of different tuning parameters on the robustness of the Markov-Laguerre based MPC. Analytical results and simulation studies were used to show that the stability properties of Markov-Laguerre based MPC and GPC are similar.

8.1.2 5. Control relevant identification for GPC

Different control-relevant ID methods due to Shook and Shah, Rivera *et al.* and Zang *et al.* were reviewed with a view to enhance the performance of GPC. The key strategy in all these control-relevant ID methods has been to estimate a control-relevant model by formulating an ID objective commensurate with the controller goal. A vital step in the implementation of this scheme has been to design a control-relevant filter by equating the control-relevant ID objective with the least-squares objective. In this context, the derivation of dufferent filters *i.e.* the L_s-filter due to Shook *et al.* , L_r-filter due to Rivera *et al.* and L_z-filter due to Zang *et al.* were reviewed. The control-relevant models were based on the filtered process data, where the above mentioned filters were used for data filtering. The idea of a frequency weighted LQG controller due to Zang *et al.* was extended to GPC for the iterative enhancement of the controller performance; and the effect of L_s, L_r and L_z filters was assessed.

The simulation results indicated that the use of an L_r filter resulted in the best performance in terms of having minimum values for the loss function along the iteration steps. The low values for the loss function were achieved by having a faster rise time but at the cost of more overshoot and by compromising the robustness margin in the mid-frequency range. The use of an L_s filter on the other hand resulted in a more uniform robustness margin which is desirable from the stability point of view but its performance was more detuned compared to the previous case. However the use of L_s filter did not give any significant overshoot due to uniform robustness margin. The use of L_z filter did not appear to be satisfactory in the context of iterative enhancement of the GPC performance.

8.2 General design and tuning guidelines for LRPC

The following steps are recommended for effective design of an LRPC:

1. Use approrpriate system identification methods to estimate a simple model from the open loop process data obtained using appropriate input excitation.

2. Use signal processing methods with appropriate smoothing techniques to estimate the spectrum of the MPM. In the case of GPC, closed loop data can also be used to carry out steps 1 and 2, while for other LRPCs, the method of determining MPM and the model from closed loop data has to be investigated further.

3. Use design methods based on the SGT to determine robust tuning parameters. Optimization methods such as SQP may also be used to determine the tuning parameters. The tuning parameters should be selected such that the robustness margin is neither too wide nor too narrow. A wide robustness margin will lead to a detuned performance whereas for a narrow margin there is a risk of violating the stability margin.

4. Implement the designed LRPC.

5. Use an iterative control relevant identification method to upgrade the process performance. In the case of GPC, the use of LRPI is recommended to enhance the process performance.

6. In the case of GPC, use batches of closed loop data to check the robustness margin from time to time. This requires mild setpoint perturbations. Consider re-tuning the controller if the characteristics of the MPM change significantly.

8.3 Future research directions

1. Robustness analysis of a multi input multi output (MIMO) system is carried out using frequency domain based singular values rather than the magnitude spectrum as was used for the SISO case. Therefore for a MIMO system the effect of how change in the norm (*i.e.* the use of singular values rather than the magnitude) would affect the conservativeness of the SGT based design. Tuning of MIMO-LRPC is another issue of interest, as it is a non-trivial task. For such a case it would be necessary to determine whether the use of the SGT can yield reliable and efficient tuning guidelines. The estimation of MPM for a MIMO system is another issue that needs to be addressed for the robust design of MIMO-LRPC. Issues such as input signal design, multivariable

spectral estimation and multivariable system identification methods would require due attention in order to estimate MPM for a MIMO system. The relative advantages and disadvantages between parametric and non-parametric signal processing methods is also an area worthy of investigation in order to estimate MPM for a MIMO system.

2. The robustness analysis in this thesis was carried out for unconstrained linear SISO systems. However in the context of robust design of LRPCs for constrained cases and mildly non-linear or saturation type non linearities, the use of methods such as conic sectors, passivity, circle theorem and Popov theorem need to be examined.

3. The robustness condition was analytically established for some of the tuning parameters of GPC for a first order model. The natural extensions of this study include: can such robustness conditions be analytically established for other tuning parameters of GPC and also for a first order model with delay; first order model with an over parameterized numerator; and also for second order models? Similarly the extensions of robustness analysis as applied to a state-space based or unstructured model based LRPC such as for Markov-Laguerre based MPC is an area worthy of further investigation.

4. A method is developed in this thesis for estimating a model and its MPM from GPC closed loop data. It would be worth exploring if such information could be extracted from closed loop data of state-space or unstructured model based LRPCs, for example, the Markov-Laguerre based MPC. In this connection it would be interesting to see if time-frequency spectral methods and wavelet spectral methods can be used to estimate the changes in plant dynamics from closed loop data of short record-lengths.

5. Control relevant identification methods need to be extended to MIMO-LRPC cases. Control relevant identification methods also need to be developed for non parametric model based LRPCs.